Appendix A

Copyright © 2000 I LOVE TV INC. (868970 Ontario Inc.)
All rights reserved unless otherwise noted

File: Ravi Shanker

```
import java.awt.*;
import java.applet.*;
import java.lang.*;
import netscape.javascript.*;
import java.io.*;
import java.net.*;
import java.util.*;

import com.oreilly.servlet.HttpMessage;

public class ChatClient extends Applet implements ISession
{
        netscape.javascript.JSObject            m_JScriptWin;

        public String                   m_strMessageHandler;
        public String                   m_strErrorHandler;
        public String                   m_EtMh;
        public String                   m_SponsorHandler;
        public String                   m_LinksHandler;
        public String                   m_AdHandler;
        public String                   m_CommentsHandler;
        public String                   m_CaptionHandler;
        public String                   m_TimeHandler           ;
        public String                   m_ClipSource;
        public String                   m_isAppletLoad;

        public Vector CurrentVector ;

        public int StartTime=0;
        public int SponsorCurrent = -1;
        public int LinkCurrent = -1;
        public int BannerCurrent = -1;
        public int CaptionCurrent = -1;
        public int CommentCurrent = -1;

        public int WhatSec = -1;

        public void init()
```

```
    {
            super.init();

    StartTime=0;
    SponsorCurrent = -1;
    LinkCurrent = -1;
    BannerCurrent = -1;
    CaptionCurrent = -1;
    CommentCurrent = -1;

    WhatSec=-1;
            // Get the JavaScript window that will have the various scripts that
            // this applet will call.
            m_JScriptWin = netscape.javascript.JSObject.getWindow(this);

            m_strMessageHandler = getParameter("MessageHandler");
            m_EtMh = getParameter("ElapsedTimeMessageHandler");
            m_strErrorHandler = getParameter("ErrorHandler");

            m_SponsorHandler = getParameter("SponsorHandler");
            m_LinksHandler = getParameter("LinksHandler");
            m_AdHandler = getParameter("AdHandler");
            m_CommentsHandler = getParameter("CommentsHandler");
            m_CaptionHandler = getParameter("CaptionHandler");
            // Set the Javascript to handle if the applet has done it's init()
            m_isAppletLoad = getParameter("isAppletLoad");

            m_TimeHandler = getParameter("TimeHandler");

            m_ClipSource = getParameter("ClipSource");

            getNextMessage(0,10,m_ClipSource);
            StartTime=10;

            if (m_JScriptWin != null)
            {
                    String Arguments[] = {"Done"};
                    m_JScriptWin.call(m_isAppletLoad, Arguments);
                    //      System.out.println("setting time here"+seconds);
            }
    }

    public void setStartTime()
    {
             int seconds = 0;
try
```

```
        {
        URL url = new URL("http://blanche.tcn.net:8090/servlet/TimeServlet");
        HttpMessage msg = new HttpMessage(url);

                Properties props = new Properties();
        props.put("ClipSource",m_ClipSource);

        InputStream in = msg.sendPostMessage(props);

        //InputStream in = msg.sendGetMessage();

        DataInputStream data = new DataInputStream(
                    new BufferedInputStream(in));
        String lastMessage = data.readLine();
        seconds  = Integer.parseInt(lastMessage);

        }
    catch (SocketException e)
    {
     // Can't connect to host, report it and wait before trying again
     System.out.println("Can't connect to host: " + e.getMessage());
    }
    catch (FileNotFoundException e)
    {
     // Servlet doesn't exist, report it and wait before trying again
     System.out.println("Resource not found: " + e.getMessage());
    }
    catch (Exception e)
    {
     // Some other problem, report it and wait before trying again
     System.out.println("General exception: " +
     e.getClass().getName() + ": " + e.getMessage());
    }
        finally
        {
                StartTime = seconds ;
                if (m_JScriptWin != null)
                        {
                                String Arguments[] = {seconds+""};
                                m_JScriptWin.call(m_TimeHandler, Arguments);
                        //      System.out.println("setting time here"+seconds);
                        }
                getNextMessage(seconds,seconds+10,m_ClipSource);
                StartTime+=10;
        }
```

```
	}

	public void start()
	{

	}

	public void stop()
	{
	}

	public int BeginSession(String strAuthor, String strOptions, String strType, String
strMessage)
	{
		return SendMessage(strAuthor, strType, strMessage);
	}

	public int EndSession(String strAuthor, String strType, String strMessage)
	{
		return SendMessage(strAuthor, strType, strMessage);
	}

	public int SendMessage(String strAuthor, String strType, String strMessage)
	{
		if (strAuthor.length() < 0)
			return HandleSessionError(1, "Invalid author specified.");

		if (strMessage.length() < 0)
			return HandleSessionError(2, "Empty message");

		if (m_strMessageHandler == null)
		{
			m_strMessageHandler = "HandleSessionMessage";
		}

		if (m_JScriptWin != null)
		{
			String Arguments[] = {strMessage};
			m_JScriptWin.call(m_strMessageHandler, Arguments);
		}

		return 0;
	}

public void SendElapsedTime(String Time)
{
```

```
                    int m_time=0;
                    try
                    {
                            m_time = Integer.parseInt(Time)/1000;
                    }
                    catch(NumberFormatException nfe)
                    {
                                    System.out.println("The Error"+nfe.toString());
                                    m_time=0;
                    }
                    //System.out.println("time sent by applet = "+m_time);
                    HandleEvent(m_time);
                    //Object Arguments[] = {(new Integer(m_time))};
                    //      m_JScriptWin.call(m_EtMh, Arguments);

                    //return 0;
    }

    public void HandleEvent(int ETime)
    {

    if(WhatSec < ETime)
    {
            Vector _Sponsor = (Vector)  CurrentVector.elementAt(0);
            //System.out.println("size of the vector is = "+_Sponsor.size()+"");
            Vector _Links = (Vector)    CurrentVector.elementAt(1);
    //System.out.println("size of the links vector is = "+_Links.size()+"");
            Vector _banner = (Vector)   CurrentVector.elementAt(2);

            Vector _caption = (Vector)  CurrentVector.elementAt(3);

            Vector _comment = (Vector) CurrentVector.elementAt(4);

            if(_Sponsor != null)
            {
                    for(int i = 0; i < _Sponsor.size();i++)
                    {
                            Vector Sp1 =(Vector) _Sponsor.elementAt(i);


                            int xy =((Integer)Sp1.elementAt(0)).intValue();
                    //System.out.println("xy = "+xy+"");
                            if(xy == ETime && SponsorCurrent != xy)
                            {
                                    if (m_JScriptWin != null)
                                    {
                                        SponsorCurrent=xy;
                                        String Arguments[] =
{Sp1.elementAt(1)+"",Sp1.elementAt(2)+"",Sp1.elementAt(3)+""};
                                        m_JScriptWin.call(m_SponsorHandler, Arguments);
                                }
                        }
                        else if(xy > ETime)
                        {
                                break;
                        }
                }// end for 1st for
        }// end of 1st if

        // for links
        if (_Links != null)
        {
                for(int i = 0; i < _Links.size();i++)
                {

                        Vector Sp2 =(Vector) _Links.elementAt(i);

                        if(Sp2 != null)
                        {
                                int xz =((Integer)Sp2.elementAt(0)).intValue();
                        //System.out.println("xy = "+xy+"");
                                if(xz == ETime)
                                {
                                        if (m_JScriptWin != null)
                                        {
                                                //System.out.println("xy = "+xz+"");
                                                String msg3="";
                                                if(     LinkCurrent == xz)
                                                {
                                                        msg3="old";
                                                }
                                                else
                                                {
                                                        LinkCurrent=xz;
                                                        msg3="new";
                                                }
                                                String Arguments[] =
{Sp2.elementAt(1)+"",Sp2.elementAt(2)+"", msg3+""};
                                                m_JScriptWin.call(m_LinksHandler,
Arguments);
                                        }
                                }
                else if(xz > ETime)
                {
                    break;
                }
            }
        }// end of 2nd for
    } // end of 1st if
        // for ads
    if (_banner != null)
    {
        for(int i = 0; i < _banner.size();i++)
        {
            Vector Sp3 =(Vector) _banner.elementAt(i);

            int az =((Integer)Sp3.elementAt(0)).intValue();
        //System.out.println("xy = "+xy+"");
            if(az == ETime && BannerCurrent != az)
            {
                if (m_JScriptWin != null)
                {
                    BannerCurrent=az;
                    String Arguments[] =
{Sp3.elementAt(1)+"",Sp3.elementAt(2)+"",Sp3.elementAt(3)+""};
                    m_JScriptWin.call(m_AdHandler, Arguments);
                }
            }
            else if(az > ETime)
            {
                break;
            }
        }
    }


    // for Caption
    if (_caption != null)
    {
        for(int i = 0; i < _caption.size();i++)
        {

            Vector Sp4 =(Vector) _caption.elementAt(i);

            if(Sp4 != null)
            {
                int nn =((Integer)Sp4.elementAt(0)).intValue();
            //System.out.println("xy = "+xy+"");
```

```
                if(nn == ETime)
                {
                    if (m_JScriptWin != null)
                    {
                        //System.out.println("xy = "+xz+"");
                        String msg3="";
                        if(     CommentCurrent == nn)
                        {
                            msg3="old";
                        }
                        else
                        {
                            CommentCurrent=nn;
                            msg3="new";
                        }
                        String Arguments[] = {Sp4.elementAt(1)+"",
msg3+""};
                        m_JScriptWin.call(m_CaptionHandler,
Arguments);
                    }
                }
                else if(nn > ETime)
                {
                    break;
                }
            }
        }// end of for
    }// end of if


    // for comment
    if (_comment != null)
    {
        for(int i = 0; i < _comment.size();i++)
        {

            Vector Sp5 =(Vector) _comment.elementAt(i);

            if(Sp5 != null)
            {
                int mm =((Integer)Sp5.elementAt(0)).intValue();
            //System.out.println("xy = "+xy+"");
                if(mm == ETime)
                {
                    if (m_JScriptWin != null)
                    {
```

```
                                        //System.out.println("xy = "+xz+"");
                                        String msg3="";
                                        if(     CaptionCurrent == mm)
                                        {
                                                msg3="old";
                                        }
                                        else
                                        {
                                                CaptionCurrent=mm;
                                                msg3="new";
                                        }
                                        String Arguments[] = {Sp5.elementAt(1)+"",
msg3+""};
                                        m_JScriptWin.call(m_CommentsHandler,
Arguments);
                                }
                        }
                        else if(mm > ETime)
                        {
                                break;
                        }
                }
        }// end of for
    }// end of if

    // if the time left for the next call is 1 mins
    if(StartTime-ETime <=1)
    {
        getNextMessage(StartTime,StartTime+10,m_ClipSource);
        StartTime=StartTime+10;
    }

    WhatSec = ETime;
}
}

public void getNextMessage(int _Start, int _End, String ClipSrc)
{
        //System.out.println("in get next message durin init");
 Object obj=null;
    try
        {
        URL url = new URL("http://blanche.tcn.net:8090/servlet/ILoveTvLiveServlet");
        //URL url = new
URL("http://torlife.wiredkingdom.com/servlet/ILoveTvLiveServlet?ElapsedTimeStart="+_St
art+"&ElapsedTimeEnd="+_End+"&ClipSource="+ClipSrc);
        HttpMessage msg = new HttpMessage(url);

                Properties props = new Properties();

       props.put("ElapsedTimeStart",_Start+"");
                props.put("ElapsedTimeEnd",_End+"");
       props.put("ClipSource",ClipSrc);

        InputStream in = msg.sendPostMessage(props);

        //InputStream in = msg.sendGetMessage();

        ObjectInputStream result = new ObjectInputStream(in);

        obj = result.readObject();
        }
    catch (SocketException e)
    {
     // Can't connect to host, report it and wait before trying again
     System.out.println("Can't connect to host: " + e.getMessage());
    }
    catch (FileNotFoundException e)
    {
     // Servlet doesn't exist, report it and wait before trying again
     System.out.println("Resource not found: " + e.getMessage());
    }
    catch (Exception e)
    {
     // Some other problem, report it and wait before trying again
     System.out.println("General exception: " +
     e.getClass().getName() + ": " + e.getMessage());
    }
  CurrentVector=(Vector)obj;
}

public int HandleSessionError(int iNumber, String strDescription)
        {
                //System.out.println(iNumber + " : " + strDescription);

                if (m_strErrorHandler == null)
                {
                        m_strErrorHandler = "HandleSessionError";
                }
        if (m_JScriptWin != null)
        {
            String Arguments[] = {Integer.toString(iNumber), strDescription};
            m_JScriptWin.call(m_strErrorHandler, Arguments);
        }

        return iNumber;
    }
}
```

09951605 .050901

Appendix B

Excalibur

File

```
VERSION 5.00
Object = "{648A5603-2C6E-101B-82B6-000000000014}#1.1#0"; "MSCOMM32.OCX"
Object = "{248DD890-BB45-11CF-9ABC-0080C7E7B78D}#1.0#0"; "MSWINSCK.OCX"
Begin VB.Form FrmLine21Watch
  Caption       =  "Line21 Watch"
  ClientHeight  =  2655
  ClientLeft    =  60
  ClientTop     =  375
  ClientWidth   =  11595
  LinkTopic     =  "Form1"
  MaxButton     =  0  'False
  MinButton     =  0  'False
  ScaleHeight   =  2655
  ScaleWidth    =  11595
  StartUpPosition =  3  'Windows Default
  Begin VB.CheckBox Field2
    Caption       =  "Field 2"
    Enabled       =  0  'False
    Height        =  255
    Left          =  9360
    TabIndex      =  7
    Top           =  2400
    Width         =  975
  End
  Begin VB.CheckBox Field1
    Caption       =  "Field 1"
    Enabled       =  0  'False
    Height        =  255
    Left          =  9360
    TabIndex      =  6
    Top           =  2040
    Width         =  855
  End
  Begin VB.CheckBox Protocol
    Caption       =  "Protocol"
    Enabled       =  0  'False
    Height        =  255
    Left          =  8160
    TabIndex      =  5
    Top           =  2040
    Width        =  975
  End
  Begin VB.ComboBox CB_Port
    Height       =  315
    Left       =  6960
    TabIndex      =  4
    Text       =  "Combo1"
    Top        =  2040
    Width        =  975
  End
  Begin VB.TextBox EB_IP
    Height       =  285
    Left       =  5400
    TabIndex      =  3
    Top         =  2040
    Width        =  1335
  End
  Begin MSWinsockLib.Winsock Winsock1
    Left       =  960
    Top         =  2040
    _ExtentX      =  741
    _ExtentY      =  741
    _Version      =  393216
  End
  Begin MSCommLib.MSComm MSComm1
    Left       =  240
    Top         =  2040
    _ExtentX      =  847
    _ExtentY      =  847
    _Version      =  393216
    DTREnable      =  -1  'True
    InputLen      =  1
    RThreshold      =  2
    SThreshold      =  1
  End
  Begin VB.CommandButton B_Watch
    Caption       =  "Watch"
    Height       =  495
    Left       =  2520
    TabIndex      =  2
    Top         =  2040
    Width        =  1455
  End
  Begin VB.ComboBox CB_ComF1
    Height       =  315
    Left       =  4320
```

```
      Style        =  2  'Dropdown List
      TabIndex       =  1
      Top          =  2040
      Width          =  855
    End
    Begin VB.ListBox LB_Triggers
      BeginProperty Font
        Name           =  "Courier New"
        Size          =  8.25
        Charset        =  0
        Weight         =  400
        Underline      =  0  'False
        Italic        =  0  'False
        Strikethrough  =  0  'False
      EndProperty
      Height         =  1740
      Left          =  120
      TabIndex       =  0
      Top            =  120
      Width          =  11175
    End
  End
  Attribute VB_Name = "FrmLine21Watch"
  Attribute VB_GlobalNameSpace = False
  Attribute VB_Creatable = False
  Attribute VB_PredeclaredId = True
  Attribute VB_Exposed = False
  Dim Loading As Boolean
  Dim running As Boolean
  Dim COMF As String
  Dim SyncFlag As Boolean
  Dim SyncCount As Integer
  Dim mes As String * 1024

  Option Explicit

  Private Sub B_Watch_Click()
    If running = False Then
      B_Watch.Caption = "Stop"
      LB_Triggers.Clear
      Line21Decode 0, 0, mes
      SyncFlag = False
      SyncCount = 0
      running = True
      If COMF <> "LAN" Then
        MSComm1.CommPort = Val(Right(COMF, 1))
```

```
   MSComm1.Settings = "9600,N,8,1"
   MSComm1.PortOpen = True
  Else
   Winsock1.RemoteHost = EB_IP.Text
   Winsock1.RemotePort = CB_Port.Text
   Winsock1.Connect
  End If
 Else
  B_Watch.Caption = "Watch"
  running = False
  If COMF <> "LAN" Then
   MSComm1.PortOpen = False
  Else
   Winsock1.Close
  End If
 End If

End Sub

Private Sub CB_ComF1_Click()
 Dim I As Integer
 Dim temp As String

 If Loading = False Then
  Loading = True
  COMF = CB_ComF1.List(CB_ComF1.ListIndex)
  If COMF = "LAN" Then
   EB_IP.Enabled = True
   CB_Port.Enabled = True
  Else
   EB_IP.Enabled = False
   CB_Port.Enabled = False
  End If
  Loading = False
 End If

End Sub

Private Sub Form_Load()
 Dim temp As String
 Dim I As Integer

 Loading = True
 running = False
 COMF = GetSetting(APPNAME, SETTING, TCOMF, "COM1")
 For I = 1 To 4
```

```
  temp = "COM" + Format(I)
  CB_ComF1.AddItem (temp)
  If COMF = temp Then
   CB_ComF1.ListIndex = CB_ComF1.ListCount - 1
  End If
 Next I
 CB_ComF1.AddItem ("LAN")
 If COMF = "LAN" Then
  CB_ComF1.ListIndex = CB_ComF1.ListCount - 1
  EB_IP.Enabled = True
  CB_Port.Enabled = True
 Else
  EB_IP.Enabled = False
  CB_Port.Enabled = False
 End If
 EB_IP.Text = GetSetting(APPNAME, SETTING, IP, "223.223.223.120")
 For I = 2000 To 9000 Step 1000
  CB_Port.AddItem (Format(I))
 Next I
 CB_Port.Text = GetSetting(APPNAME, SETTING, PORT, "3000")
 Loading = False

End Sub

Private Sub Form_Unload(Cancel As Integer)
 SaveSetting APPNAME, SETTING, TCOMF, COMF
 SaveSetting APPNAME, SETTING, IP, EB_IP.Text
 SaveSetting APPNAME, SETTING, PORT, CB_Port.Text

End Sub

Private Sub MSComm1_OnComm()
 Dim Trig As Long
 Dim status As Integer
 Static Num As Integer
 Static Dat As String
 Dim C0 As Long
 Dim C1 As Long
 Dim C2 As Long

 status = MSComm1.CommEvent
 If status = 2 And running = True Then
  If SyncFlag = False Then          'Getting is sync is real ugly
   SyncCount = SyncCount + 1         'We watch for the protocol characters
   If SyncCount = 1 Then
    MSComm1.InBufferCount = 0
 Else
  Dat = MSComm1.Input           'and keep track of whether data is from
  Select Case Asc(Dat)          'field 1, field 2 or both.
   Case 17, 129                 'We want to start capturing data on field 1
    Protocol.Value = 1          'if both fields or at the start of the
    If Field1.Value = 0 And Field2.Value = 1 Then
     Num = 3
    Else
     Num = 0
    End If
    Field1.Value = 1            'field data.
   Case 18, 130
    Protocol.Value = 1
    Num = 0
    Field2.Value = 1
   Case 148, 28                 'If it is not protocol data then we must
    If SyncCount > 2 And Protocol.Value = 0 Then
     Field1.Value = 1           'decide on the field based on the control
     Num = 1                    'codes.
    End If
   Case 21, 157, 1, 2, 131, 4, 133, 134, 7, 8, 137, 138, 11, 140, 13, 14, 143
    If SyncCount > 2 And Protocol.Value = 0 Then
     Field2.Value = 1
     Num = 1
    End If
   Case Else
    Num = Num + 1
  End Select
  If SyncCount > 5 And (Field1.Value = 1 Or Field2.Value = 1) And Num = 2 Then
   SyncFlag = True
   If Protocol.Value = 1 Then
    MSComm1.InputLen = 3
   Else
    MSComm1.InputLen = 2
   End If
  End If
 End If
Else
 Dat = MSComm1.Input
 If Protocol.Value = 1 Then
  C0 = CByte(Asc(Left(Dat, 1)))
  C1 = CByte(Asc(Mid(Dat, 2, 1))) And 127
  C2 = CByte(Asc(Right(Dat, 1))) And 127
 Else
  If Field1.Value = 1 Then
   C0 = 129
```

```
      Else
        C0 = 130
      End If
    End If
    Select Case C0
      Case 129, 17
        Trig = Line21Decode(1, C1, mes)
        If Mid(mes, 1, 1) <> Chr(0) Then
          LB_Triggers.AddItem (mes)
        End If
        Trig = Line21Decode(1, C2, mes)
        If Mid(mes, 1, 1) <> Chr(0) Then
          LB_Triggers.AddItem (mes)
        End If
      Case 130, 18
        Trig = Line21Decode(2, C1, mes)
        If Mid(mes, 1, 1) <> Chr(0) Then
          LB_Triggers.AddItem (mes)
        End If
        Trig = Line21Decode(2, C2, mes)
        If Mid(mes, 1, 1) <> Chr(0) Then
          LB_Triggers.AddItem (mes)
        End If
    End Select
  End If
 End If
End Sub


Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
 Static InputLen As Integer
 Static Num As Integer
 Static Dat As String
 Dim Trig As Long
 Dim C0 As Long
 Dim C1 As Long
 Dim C2 As Long

 If running = True Then
   If InputLen < 1 Or InputLen > 3 Then
     InputLen = 1
   End If
   While Winsock1.BytesReceived >= InputLen
     If SyncFlag = False Then          'Getting in sync is real ugly
       SyncCount = SyncCount + 1          'We watch for the protocol characters
       If SyncCount = 1 Then
 MSComm1.InBufferCount = 0
Else
 Winsock1.GetData Dat, , 1      'and keep track of whether data is from
 Select Case Asc(Dat)        'field 1, field 2 or both.
  Case 17, 129            'We want to start capturing data on field 1
   Protocol.Value = 1        'if both fields or at the start of the
   If Field1.Value = 0 And Field2.Value = 1 Then
    Num = 3
   Else
    Num = 0
   End If
   Field1.Value = 1           'field data.
  Case 18, 130
   Protocol.Value = 1
   Num = 0
   Field2.Value = 1
  Case 148, 28              'If it is not protocol data then we must
   If SyncCount > 2 And Protocol.Value = 0 Then
    Field1.Value = 1          'decide on the field based on the control
    Num = 1                'codes.
   End If
  Case 21, 157, 1, 2, 131, 4, 133, 134, 7, 8, 137, 138, 11, 140, 13, 14, 143
   If SyncCount > 2 And Protocol.Value = 0 Then
    Field2.Value = 1
    Num = 1
   End If
  Case Else
   Num = Num + 1
 End Select
 If SyncCount > 5 And (Field1.Value = 1 Or Field2.Value = 1) And Num = 2 Then
  SyncFlag = True
  If Protocol.Value = 1 Then
   InputLen = 3
  Else
   InputLen = 2
  End If
 End If
End If
Else
 Winsock1.GetData Dat, , InputLen
 If Protocol.Value = 1 Then
  C0 = CByte(Asc(Left(Dat, 1)))
  C1 = CByte(Asc(Mid(Dat, 2, 1))) And 127
  C2 = CByte(Asc(Right(Dat, 1))) And 127
 Else
  If Field1.Value = 1 Then
```

```
      C0 = 129
     Else
      C0 = 130
     End If
    End If
    Select Case C0
     Case 129, 17
      Trig = Line21Decode(1, C1, mes)
      If Mid(mes, 1, 1) <> Chr(0) Then
       LB_Triggers.AddItem (mes)
      End If
      Trig = Line21Decode(1, C2, mes)
      If Mid(mes, 1, 1) <> Chr(0) Then
       LB_Triggers.AddItem (mes)
      End If
     Case 130, 18
      Trig = Line21Decode(2, C1, mes)
      If Mid(mes, 1, 1) <> Chr(0) Then
       LB_Triggers.AddItem (mes)
      End If
      Trig = Line21Decode(2, C2, mes)
      If Mid(mes, 1, 1) <> Chr(0) Then
       LB_Triggers.AddItem (mes)
      End If
    End Select
   End If
  Wend
 End If

End Sub
```

Appendix B

Copyright © 2000 I LOVE TV INC. (868970 Ontario Inc.)
All rights reserved

Excalibur

File: FrmLine21Watch.frm

```
VERSION 5.00
Object = "{648A5603-2C6E-101B-82B6-000000000014}#1.1#0"; "MSCOMM32.OCX"
Object = "{248DD890-BB45-11CF-9ABC-0080C7E7B78D}#1.0#0"; "MSWINSCK.OCX"
Begin VB.Form FrmLine21Watch
  Caption        =   "Line21 Watch"
  ClientHeight   =   2655
  ClientLeft     =   60
  ClientTop      =   375
  ClientWidth    =   11595
  LinkTopic      =   "Form1"
  MaxButton      =   0   'False
  MinButton      =   0   'False
  ScaleHeight    =   2655
  ScaleWidth     =   11595
  StartUpPosition =  3   'Windows Default
  Begin VB.CheckBox Field2
    Caption        =   "Field 2"
    Enabled        =   0   'False
    Height         =   255
    Left           =   9360
    TabIndex       =   7
    Top            =   2400
    Width          =   975
  End
  Begin VB.CheckBox Field1
    Caption        =   "Field 1"
    Enabled        =   0   'False
    Height         =   255
    Left           =   9360
    TabIndex       =   6
    Top            =   2040
    Width          =   855
  End
  Begin VB.CheckBox Protocol
    Caption        =   "Protocol"
    Enabled        =   0   'False
    Height         =   255
```

```
   Left          =  8160
   TabIndex      =  5
   Top           =  2040
   Width         =  975
End
Begin VB.ComboBox CB_Port
   Height        =  315
   Left          =  6960
   TabIndex      =  4
   Text          =  "Combo1"
   Top           =  2040
   Width         =  975
End
Begin VB.TextBox EB_IP
   Height        =  285
   Left          =  5400
   TabIndex      =  3
   Top           =  2040
   Width         =  1335
End
Begin MSWinsockLib.Winsock Winsock1
   Left          =  960
   Top           =  2040
   _ExtentX      =  741
   _ExtentY      =  741
   _Version      =  393216
End
Begin MSCommLib.MSComm MSComm1
   Left          =  240
   Top           =  2040
   _ExtentX      =  847
   _ExtentY      =  847
   _Version      =  393216
   DTREnable     =  -1 'True
   InputLen      =  1
   RThreshold    =  2
   SThreshold    =  1
End
Begin VB.CommandButton B_Watch
   Caption       =  "Watch"
   Height        =  495
   Left          =  2520
   TabIndex      =  2
   Top           =  2040
   Width         =  1455
End
```

```
  Begin VB.ComboBox CB_ComF1
    Height          =   315
    Left            =   4320
    Style           =   2  'Dropdown List
    TabIndex        =   1
    Top             =   2040
    Width           =   855
  End
  Begin VB.ListBox LB_Triggers
    BeginProperty Font
      Name            =   "Courier New"
      Size            =   8.25
      Charset         =   0
      Weight          =   400
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
    EndProperty
    Height          =   1740
    Left            =   120
    TabIndex        =   0
    Top             =   120
    Width           =   11175
  End
End
Attribute VB_Name = "FrmLine21Watch"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Dim Loading As Boolean
Dim running As Boolean
Dim COMF As String
Dim SyncFlag As Boolean
Dim SyncCount As Integer
Dim mes As String * 1024

Option Explicit

Private Sub B_Watch_Click()
 If running = False Then
   B_Watch.Caption = "Stop"
   LB_Triggers.Clear
   Line21Decode 0, 0, mes
   SyncFlag = False
   SyncCount = 0
  running = True
  If COMF <> "LAN" Then
   MSComm1.CommPort = Val(Right(COMF, 1))
   MSComm1.Settings = "9600,N,8,1"
   MSComm1.PortOpen = True
  Else
   Winsock1.RemoteHost = EB_IP.Text
   Winsock1.RemotePort = CB_Port.Text
   Winsock1.Connect
  End If
 Else
  B_Watch.Caption = "Watch"
  running = False
  If COMF <> "LAN" Then
   MSComm1.PortOpen = False
  Else
   Winsock1.Close
  End If
 End If

End Sub

Private Sub CB_ComF1_Click()
 Dim I As Integer
 Dim temp As String

 If Loading = False Then
  Loading = True
  COMF = CB_ComF1.List(CB_ComF1.ListIndex)
  If COMF = "LAN" Then
   EB_IP.Enabled = True
   CB_Port.Enabled = True
  Else
   EB_IP.Enabled = False
   CB_Port.Enabled = False
  End If
  Loading = False
 End If

End Sub

Private Sub Form_Load()
 Dim temp As String
 Dim I As Integer

 Loading = True
```

```
running = False
COMF = GetSetting(APPNAME, SETTING, TCOMF, "COM1")
For I = 1 To 4
 temp = "COM" + Format(I)
 CB_ComF1.AddItem (temp)
 If COMF = temp Then
  CB_ComF1.ListIndex = CB_ComF1.ListCount - 1
 End If
Next I
CB_ComF1.AddItem ("LAN")
If COMF = "LAN" Then
 CB_ComF1.ListIndex = CB_ComF1.ListCount - 1
 EB_IP.Enabled = True
 CB_Port.Enabled = True
Else
 EB_IP.Enabled = False
 CB_Port.Enabled = False
End If
EB_IP.Text = GetSetting(APPNAME, SETTING, IP, "223.223.223.120")
For I = 2000 To 9000 Step 1000
 CB_Port.AddItem (Format(I))
Next I
CB_Port.Text = GetSetting(APPNAME, SETTING, PORT, "3000")
Loading = False

End Sub

Private Sub Form_Unload(Cancel As Integer)
 SaveSetting APPNAME, SETTING, TCOMF, COMF
 SaveSetting APPNAME, SETTING, IP, EB_IP.Text
 SaveSetting APPNAME, SETTING, PORT, CB_Port.Text

End Sub

Private Sub MSComm1_OnComm()
 Dim Trig As Long
 Dim status As Integer
 Static Num As Integer
 Static Dat As String
 Dim C0 As Long
 Dim C1 As Long
 Dim C2 As Long

 status = MSComm1.CommEvent
 If status = 2 And running = True Then
  If SyncFlag = False Then          'Getting is sync is real ugly
```

```
SyncCount = SyncCount + 1        'We watch for the protocol characters
If SyncCount = 1 Then
 MSComm1.InBufferCount = 0
Else
 Dat = MSComm1.Input             'and keep track of whether data is from
 Select Case Asc(Dat)            'field 1, field 2 or both.
  Case 17, 129                   'We want to start capturing data on field 1
   Protocol.Value = 1            'if both fields or at the start of the
   If Field1.Value = 0 And Field2.Value = 1 Then
    Num = 3
   Else
    Num = 0
   End If
   Field1.Value = 1              'field data.
  Case 18, 130
   Protocol.Value = 1
   Num = 0
   Field2.Value = 1
  Case 148, 28                   'If it is not protocol data then we must
   If SyncCount > 2 And Protocol.Value = 0 Then
    Field1.Value = 1             'decide on the field based on the control
    Num = 1                      'codes.
   End If
  Case 21, 157, 1, 2, 131, 4, 133, 134, 7, 8, 137, 138, 11, 140, 13, 14, 143
   If SyncCount > 2 And Protocol.Value = 0 Then
    Field2.Value = 1
    Num = 1
   End If
  Case Else
   Num = Num + 1
 End Select
 If SyncCount > 5 And (Field1.Value = 1 Or Field2.Value = 1) And Num = 2 Then
  SyncFlag = True
  If Protocol.Value = 1 Then
   MSComm1.InputLen = 3
  Else
   MSComm1.InputLen = 2
  End If
 End If
End If
Else
 Dat = MSComm1.Input
 If Protocol.Value = 1 Then
  C0 = CByte(Asc(Left(Dat, 1)))
  C1 = CByte(Asc(Mid(Dat, 2, 1))) And 127
  C2 = CByte(Asc(Right(Dat, 1))) And 127
```

```
      Else
       If Field1.Value = 1 Then
        C0 = 129
       Else
        C0 = 130
       End If
      End If
      Select Case C0
       Case 129, 17
        Trig = Line21Decode(1, C1, mes)
        If Mid(mes, 1, 1) <> Chr(0) Then
         LB_Triggers.AddItem (mes)
        End If
        Trig = Line21Decode(1, C2, mes)
        If Mid(mes, 1, 1) <> Chr(0) Then
         LB_Triggers.AddItem (mes)
        End If
       Case 130, 18
        Trig = Line21Decode(2, C1, mes)
        If Mid(mes, 1, 1) <> Chr(0) Then
         LB_Triggers.AddItem (mes)
        End If
        Trig = Line21Decode(2, C2, mes)
        If Mid(mes, 1, 1) <> Chr(0) Then
         LB_Triggers.AddItem (mes)
        End If
      End Select
     End If
    End If
End Sub


Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
  Static InputLen As Integer
  Static Num As Integer
  Static Dat As String
  Dim Trig As Long
  Dim C0 As Long
  Dim C1 As Long
  Dim C2 As Long

  If running = True Then
   If InputLen < 1 Or InputLen > 3 Then
    InputLen = 1
   End If
   While Winsock1.BytesReceived >= InputLen
```

```
If SyncFlag = False Then          'Getting in sync is real ugly
 SyncCount = SyncCount + 1          'We watch for the protocol characters
 If SyncCount = 1 Then
  MSComm1.InBufferCount = 0
 Else
  Winsock1.GetData Dat, , 1        'and keep track of whether data is from
  Select Case Asc(Dat)          'field 1, field 2 or both.
   Case 17, 129          'We want to start capturing data on field 1
    Protocol.Value = 1          'if both fields or at the start of the
    If Field1.Value = 0 And Field2.Value = 1 Then
     Num = 3
    Else
     Num = 0
    End If
    Field1.Value = 1          'field data.
   Case 18, 130
    Protocol.Value = 1
    Num = 0
    Field2.Value = 1
   Case 148, 28          'If it is not protocol data then we must
    If SyncCount > 2 And Protocol.Value = 0 Then
     Field1.Value = 1          'decide on the field based on the control
     Num = 1          'codes.
    End If
   Case 21, 157, 1, 2, 131, 4, 133, 134, 7, 8, 137, 138, 11, 140, 13, 14, 143
    If SyncCount > 2 And Protocol.Value = 0 Then
     Field2.Value = 1
     Num = 1
    End If
   Case Else
    Num = Num + 1
  End Select
  If SyncCount > 5 And (Field1.Value = 1 Or Field2.Value = 1) And Num = 2 Then
   SyncFlag = True
   If Protocol.Value = 1 Then
    InputLen = 3
   Else
    InputLen = 2
   End If
  End If
 End If
Else
 Winsock1.GetData Dat, , InputLen
 If Protocol.Value = 1 Then
  C0 = CByte(Asc(Left(Dat, 1)))
  C1 = CByte(Asc(Mid(Dat, 2, 1))) And 127
```

```
    C2 = CByte(Asc(Right(Dat, 1))) And 127
   Else
    If Field1.Value = 1 Then
     C0 = 129
    Else
     C0 = 130
    End If
   End If
   Select Case C0
    Case 129, 17
     Trig = Line21Decode(1, C1, mes)
     If Mid(mes, 1, 1) <> Chr(0) Then
      LB_Triggers.AddItem (mes)
     End If
     Trig = Line21Decode(1, C2, mes)
     If Mid(mes, 1, 1) <> Chr(0) Then
      LB_Triggers.AddItem (mes)
     End If
    Case 130, 18
     Trig = Line21Decode(2, C1, mes)
     If Mid(mes, 1, 1) <> Chr(0) Then
      LB_Triggers.AddItem (mes)
     End If
     Trig = Line21Decode(2, C2, mes)
     If Mid(mes, 1, 1) <> Chr(0) Then
      LB_Triggers.AddItem (mes)
     End If
   End Select
   End If
  Wend
 End If

End Sub
```

File: FrmLine21Watch.log

```
Line 15: Class MSCommLib.MSComm of control MSComm1 was not a loaded control class.
Line 18: The property name _ExtentX in MSComm1 is invalid.
Line 19: The property name _ExtentY in MSComm1 is invalid.
Line 20: The property name _Version in MSComm1 is invalid.
Line 21: The property name DTREnable in MSComm1 is invalid.
Line 22: The property name RThreshold in MSComm1 is invalid.
```

File: Funt.c

Copyright © 2000 I LOVE TV INC. (868970 Ontario Inc.)
All rights reserved

```
#include <windows.h>
#include <stdio.h>
#include <string.h>
#include "L21Dec.h"
#include "sMachine.h"

extern char MaxLength[CLASSMAX][RECORDMAX];
extern char MinLength[CLASSMAX][RECORDMAX];
extern char XDSMes[CLASSMAX][RECORDMAX][MESLEN];
extern void (*XDSDisp[CLASSMAX][RECORDMAX])();
extern char *CType[];

// When we start an XDS record we must remember which class it is from

int XDSBegin(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
  {
  cStatePtr->XDSClass = (ch - 1) / 2;
  return(TRUE);
  }

// We art starting the XDS record so we must remember the record.  If the record is
// already open then we should give an overwrite error.  Prepare the XDS data structure
// for the record.

int XDSStart(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
  {
  cStatePtr->XDSRecord = ch;
  if(cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].Open == TRUE)
    {
    strcpy(mes,XDSSTART);
    strcat(mes,XDSMes[cStatePtr->XDSClass][cStatePtr->XDSRecord]);
    strcat(mes,OVERWRITE);
    }
  cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].Open = TRUE;
  cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].Length = 0;
  cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].CheckSum = cStatePtr-
>XDSClass * 2 + 1 + ch;
  return(TRUE);
```

```
}

// When we continue an XDS record we must save the record type.  We check to see if the
// record is open.  If it is not then we give an error message and open it as if this
// was a start.

int XDSCont(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
 {
 cStatePtr->XDSRecord = ch;
 if(cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].Open == FALSE)
   {
   strcpy(mes,XDSSTART);
   strcat(mes,XDSMes[cStatePtr->XDSClass][cStatePtr->XDSRecord]);
   strcat(mes,CONTNOSTART);
   cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].Open = TRUE;
   cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].Length = 0;
   cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].CheckSum = cStatePtr-
>XDSClass * 2 + 1 + ch;
   }
  return(TRUE);
  }

// When a character arrives for the XDS record we must add it to the checksum.  We check
// to see if this makes the record longer than the max length for this record.  If not
// then we store the data for this record and increment the record length.

int XDSData(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
  {
  cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].CheckSum += ch;
  if(cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].Length <
MaxLength[cStatePtr->XDSClass][cStatePtr->XDSRecord])
    cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].Data[cStatePtr-
>XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].Length] = ch;
  cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].Length++;
  return(TRUE);
  }

// When we close an XDS record we must check to see that we have received enough data,
// that we have not received too much data and that the checksum is valid.  Any of these
// will cause error messages.  If the record is OK then display the XDS data.  This is
// done through a look-up table of routines so each XDS record can be displayed
// differently.

int XDSEnd(long field, unsigned char ch, char mes[],  struct cState *cStatePtr)
  {
  cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].Open = FALSE;
  if(cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].Length <
MinLength[cStatePtr->XDSClass][cStatePtr->XDSRecord])
    {
    strcpy(mes,XDSSTART);
    strcat(mes,XDSMes[cStatePtr->XDSClass][cStatePtr->XDSRecord]);
    strcat(mes,TOOSHORT);
    }
  else if(cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].Length >
MaxLength[cStatePtr->XDSClass][cStatePtr->XDSRecord])
    {
    strcpy(mes,XDSSTART);
    strcat(mes,XDSMes[cStatePtr->XDSClass][cStatePtr->XDSRecord]);
    strcat(mes,TOOLONG);
    }
  else
    {
    if(((cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].CheckSum +
XDSEND + ch) & 0x7f) != 0)
      {
      strcpy(mes,XDSSTART);
      strcat(mes,XDSMes[cStatePtr->XDSClass][cStatePtr->XDSRecord]);
      strcat(mes,BADCHECKSUM);
      }
    else
      XDSDisp[cStatePtr->XDSClass][cStatePtr->XDSRecord](mes, cStatePtr);
    }
  return(TRUE);
  }

// When we get a control character (0x10 to 0x1f) we must save it so we can use
// it as part of a control pair with the next character.

int ControlSave(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
  {
  cStatePtr->CurrentCtrl[field - 1] = ch;
  return(TRUE);
  }

// For each of the four possible streams for each field we must maintain a buffer of
// the characters that have arrived for the stream.  We will detect the "end of a caption
// and dump the caption as a message.

int DataSave(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
  {
  int f, c, m;
```

```
f = field - 1;
c = cStatePtr->Channel[f];
m = cStatePtr->Mode[f];
if(cStatePtr->TextLength[f][c][m] < 255)
  {
  cStatePtr->Text[f][c][m][cStatePtr->TextLength[f][c][m]] = ch;
  cStatePtr->TextLength[f][c][m]++;
  }
return(TRUE);
}

// There are different types of captions.  We store the type of caption for the stream.

int PopOnDecode(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
 {
 int f;

 f = field - 1;
 cStatePtr->Mode[f] = CAP;
 if(cStatePtr->CurrentCtrl[f] < 0x18)
   {
   cStatePtr->Channel[f] = C1;
   cStatePtr->Type[f][C1] = POPUP;
   }
 else
   {
   cStatePtr->Channel[f] = C2;
   cStatePtr->Type[f][C2] = POPUP;
   }
 return(TRUE);
 }

int RollUpDecode(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
 {
 int f, c;

 f = field - 1;
 cStatePtr->Mode[f] = CAP;
 if(cStatePtr->CurrentCtrl[f] < 0x18)
   c = C1;
 else
   c = C2;
 cStatePtr->Channel[f] = c;
 if(ch == 0x25)
   cStatePtr->Type[f][c] = ROLLUP2;
 if(ch == 0x26)
```

```
   cStatePtr->Type[f][c] = ROLLUP3;
  if(ch == 0x27)
   cStatePtr->Type[f][c] = ROLLUP4;
  return(TRUE);
  }

int PaintOnDecode(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
  {
  int f;

  f = field - 1;
  cStatePtr->Mode[f] = CAP;
  if(cStatePtr->CurrentCtrl[f] < 0x18)
    {
    cStatePtr->Channel[f] = C1;
    cStatePtr->Type[f][C1] = PAINTON;
    }
  else
    {
    cStatePtr->Channel[f] = C2;
    cStatePtr->Type[f][C2] = PAINTON;
    }
  return(TRUE);
  }

int EndOfCaptionDecode(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
  {
  int f;

  f = field - 1;
  cStatePtr->Mode[f] = CAP;
  if(cStatePtr->CurrentCtrl[f] < 0x18)
    {
    cStatePtr->Channel[f] = C1;
    cStatePtr->Type[f][C1] = POPUP;
    }
  else
    {
    cStatePtr->Channel[f] = C2;
    cStatePtr->Type[f][C2] = POPUP;
    }
  return(TRUE);
  }

// If the stream is not a caption stream then it is a text stream and should be marked
// as such.  The type does not applie to the stream if it is a text stream.
```

```
int TextDecode(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
 {
 int f;

 f = field - 1;
 cStatePtr->Mode[f] = TXT;
 if(cStatePtr->CurrentCtrl[f] < 0x18)
   cStatePtr->Channel[f] = C1;
 else
   cStatePtr->Channel[f] = C2;
 return(TRUE);
 }

// URLs are a special use of the T2 stream.  We track the T2 stream outside of the
// regular text streams so that we can monitor it for the URL checksum which we use
// to determine the end of the URL.

int URLDecode(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
 {
 cStatePtr->Mode[0] = TXT;
 cStatePtr->Channel[0] = C2;
 return(TRUE);
 }

// This is a tricky one.  It is possible to change streams with a mid row code.  In this
// case we change where we put text.

int ChannelChange(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
 {
 int f;

 f = field - 1;
 if((ch >= 0x20) && (ch <= 0x3f))
   {
   if(cStatePtr->CurrentCtrl[f] < 0x18)
     cStatePtr->Channel[f] = C1;
   else
     cStatePtr->Channel[f] = C2;
   }
 return(TRUE);
 }

// This is where we figure out what the current stream is an create a message which
// includes the stream type, the caption type (if applicable) and the text we have
// captured for this stream.  We will call this if:  we get a control pair carriage
// return, we get a possitioning code pair (PAC) or we see a checksum in the URL
// stream.

int DumpText(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
 {
 int f, c, m;

 f = field - 1;
 m = cStatePtr->Mode[f];
 c = cStatePtr->Channel[f];
 if(cStatePtr->TextLength[f][c][m] > 0)
  {
  if(field == FIELD1)
   {
   if(m == CAP)
    {
    if(c == C1)
     {
     strcpy(mes, C1START);
     strcat(mes, CType[cStatePtr->Type[f][c]]);
     }
    else
     {
     strcpy(mes, C2START);
     strcat(mes, CType[cStatePtr->Type[f][c]]);
     }
    }
   else
    {
    if(c == C1)
     {
     strcpy(mes, T1START);
     strcat(mes, SEVENSP);
     }
    else
     {
     strcpy(mes, T2START);
     strcat(mes, SEVENSP);
     }
    }
   }
  else
   {
   if(m == CAP)
    {
    if(c == C1)
    {
    strcpy(mes, C3START);
    strcat(mes, CType[cStatePtr->Type[f][c]]);
    }
   else
    {
    strcpy(mes, C4START);
    strcat(mes, CType[cStatePtr->Type[f][c]]);
    }
   }
  else
   {
   if(c == C1)
    {
    strcpy(mes, T3START);
    strcat(mes, SEVENSP);
    }
   else
    {
    strcpy(mes, T4START);
    strcat(mes, SEVENSP);
    }
   }
  }
 cStatePtr->Text[f][c][m][cStatePtr->TextLength[f][c][m]] = 0;
 strcat(mes, cStatePtr->Text[f][c][m]);
 cStatePtr->TextLength[f][c][m] = 0;
 }
return(TRUE);
}
```

File: L21Dec.c

Copyright © 2000 I LOVE TV INC. (868970 Ontario Inc.)
All rights reserved

```
#include "windows.h"
#include "stdio.h"
#include "L21Dec.h"
#include "sMachine.h"

extern SVAL DataState[];
extern unsigned char parity[];

struct cState CCstate;
struct cState *cStatePtr;

#pragma warning(disable : 4133)

BOOL APIENTRY DllMain(HANDLE hInst, DWORD ul_reason_being_called, LPVOID
lpReserved)
 {
 return 1;
 UNREFERENCED_PARAMETER(hInst);
 UNREFERENCED_PARAMETER(ul_reason_being_called);
 UNREFERENCED_PARAMETER(lpReserved);
 }

__declspec(dllexport) LONG APIENTRY Line21Decode(long field, unsigned char ch, char
mes[])
 {
 int i, j;

 if(field == 0)  // Reset the finate state machine
   {
   cStatePtr = &CCstate;
   cStatePtr->NextState[0] = DataState;
   cStatePtr->NextState[1] = DataState;
   cStatePtr->Channel[0] = C1;
   cStatePtr->Channel[1] = C1;
   cStatePtr->Mode[0] = CAP;
   cStatePtr->Mode[1] = CAP;
   for(i = 0;i < CLASSMAX;i++)
     {
     for(j = 0;j < RECORDMAX;j++)
```

```
      cStatePtr->XDS[i][j].Open = FALSE;
     }
    }
  else if((field == 2) || (field == 1))
    {
    ch &= 0x7f;
    mes[0] = 0;
    stateMachine(field, ch, mes);
    }
  return(0);
  }


void stateMachine(long field, unsigned char ch, char mes[])
  {
  SVAL *cstate;

/*
 *  This is the state machine driver.  Each possible branch of a state
 *  starts with a test.  If the test is passed then the functions of that
 *  branch are executed.  The end of functions is indicated by a NULL
 *  pointer.  After the NULL pointer is a pointer to the next state.  The
 *  whole state is a single array containing sets of, the state test,
 *  the functions for that state branch and the next state.  The test of the last
 *  set must be the default test which always returns true.
 */
  cstate = cStatePtr->NextState[field - 1];
  while(TRUE)
    {
    if(cstate++->fptr(field, ch, mes, cStatePtr))   // Do the test for this branch
      {
      while(*cstate->fptr)             // Passed so execute all its functions
        {
        cstate++->fptr(field, ch, mes, cStatePtr);
        }
      cStatePtr->NextState[field - 1] = (cstate +1)->dptr;  // Then return pointer to the next state
      break;
      }
    else                            // Failed this branch test so
      {
      while(*cstate++->fptr)            // Skip over its functions
        ;
      cstate++;                        // Skip the state change to the
      }                              //  start of the next branch set
    }
  }
```

```
__declspec(dllexport) LONG APIENTRY ShiftRight(unsigned long value, long shift)
{
return(value >> shift);
}

__declspec(dllexport) LONG APIENTRY ShiftLeft(unsigned long value, long shift)
{
return(value << shift);
}
```

09654505 .050901

File: L21Dec.def

```
LIBRARY   Line21

CODE   PRELOAD MOVEABLE DISCARDABLE
DATA   PRELOAD SINGLE

EXPORTS
   Line21Decode
   ShiftRight
         ShiftLeft
```

09851605.050901

File: L21Dec.h

```
__declspec(dllexport) LONG APIENTRY Line21Decode(long field, unsigned char ch, char
mess[]);
__declspec(dllexport) LONG APIENTRY ShiftRight(unsigned long, long);
__declspec(dllexport) LONG APIENTRY ShiftLeft(unsigned long, long);

void stateMachine(long field, unsigned char ch, char mes[]);
void Def(char mes[], struct cState *);
void Vcp(char mes[], struct cState *);
void Typ(char mes[], struct cState *);

int XDSBegin(long field, unsigned char ch, char mes[], struct cState *);
int XDSStart(long field, unsigned char ch, char mes[], struct cState *);
int XDSCont(long field, unsigned char ch, char mes[], struct cState *);
int XDSStart(long field, unsigned char ch, char mes[], struct cState *);
int XDSData(long field, unsigned char ch, char mes[], struct cState *);
int XDSEnd(long field, unsigned char ch, char mes[], struct cState *);
int ChannelChange(long field, unsigned char ch, char mes[], struct cState *);
int DumpText(long field, unsigned char ch, char mes[], struct cState *);
int ControlSave(long field, unsigned char ch, char mes[], struct cState *);
int PopOnDecode(long field, unsigned char ch, char mes[], struct cState *);
int DataSave(long field, unsigned char ch, char mes[], struct cState *);
int RollUpDecode(long field, unsigned char ch, char mes[], struct cState *);
int PaintOnDecode(long field, unsigned char ch, char mes[], struct cState *);
int TextDecode(long field, unsigned char ch, char mes[], struct cState *);
int URLDecode(long field, unsigned char ch, char mes[], struct cState *);

int DefaultTest(long field, unsigned char ch, char mes[], struct cState *);
int NullTest(long field, unsigned char ch, char mes[], struct cState *);
int F1XDSTest(long field, unsigned char ch, char mes[], struct cState *);
int ControlTest(long field, unsigned char ch, char mes[], struct cState *);
int XDSStartTest(long field, unsigned char ch, char mes[], struct cState *);
int XDSContTest(long field, unsigned char ch, char mes[], struct cState *);
int XDSEndTest(long field, unsigned char ch, char mes[], struct cState *);
int RollUpTest(long field, unsigned char ch, char mes[], struct cState *);
int PopOnTest(long field, unsigned char ch, char mes[], struct cState *);
int PaintOnTest(long field, unsigned char ch, char mes[], struct cState *);
int EndOfCaptionTest(long field, unsigned char ch, char mes[], struct cState *);
int TextModeTest(long field, unsigned char ch, char mes[], struct cState *);
int PACTest(long field, unsigned char ch, char mes[], struct cState *);
int CRTest(long field, unsigned char ch, char mes[], struct cState *);
int CtrlChangeTest(long field, unsigned char ch, char mes[], struct cState *);
int URLTest(long field, unsigned char ch, char mes[], struct cState *);
int URLEndTest(long field, unsigned char ch, char mes[], struct cState *);
```

File: Line21.mak

```
# Microsoft Developer Studio Generated NMAKE File, Format Version 40001
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

!IF "$(CFG)" == ""
CFG=Line21 - Win32 Debug
!MESSAGE No configuration specified.  Defaulting to Line21 - Win32 Debug.
!ENDIF

!IF "$(CFG)" != "Line21 - Win32 Release" && "$(CFG)" != "Line21 - Win32 Debug"
!MESSAGE Invalid configuration "$(CFG)" specified.
!MESSAGE You can specify a configuration when running NMAKE on this makefile
!MESSAGE by defining the macro CFG on the command line.  For example:
!MESSAGE
!MESSAGE NMAKE /f "Line21.mak" CFG="Line21 - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "Line21 - Win32 Release" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "Line21 - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE
!ERROR An invalid configuration is specified.
!ENDIF

!IF "$(OS)" == "Windows_NT"
NULL=
!ELSE
NULL=nul
!ENDIF
###########################################################################
#####
# Begin Project
# PROP Target_Last_Scanned "Line21 - Win32 Debug"
RSC=rc.exe
MTL=mktyplib.exe
CPP=cl.exe

!IF  "$(CFG)" == "Line21 - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir "Release"
# PROP Intermediate_Dir "Release"
# PROP Target_Dir ""
OUTDIR=.\Release
INTDIR=.\Release

ALL : "$(OUTDIR)\Line21.dll"

CLEAN :
        -@erase ".\Release\Line21.dll"
        -@erase ".\Release\XDSROM.obj"
        -@erase ".\Release\stable.obj"
        -@erase ".\Release\L21Dec.obj"
        -@erase ".\Release\XDSMessages.obj"
        -@erase ".\Release\Tests.obj"
        -@erase ".\Release\funct.obj"
        -@erase ".\Release\Parity.obj"
        -@erase ".\Release\Line21.lib"
        -@erase ".\Release\Line21.exp"

"$(OUTDIR)" :
   if not exist "$(OUTDIR)/$(NULL)" mkdir "$(OUTDIR)"

# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D
"_WINDOWS" /YX /c
# ADD CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX
/c
CPP_PROJ=/nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS"\
 /Fp"$(INTDIR)/Line21.pch" /YX /Fo"$(INTDIR)/" /c
CPP_OBJS=.\Release/
CPP_SBRS=
# ADD BASE MTL /nologo /D "NDEBUG" /win32
# ADD MTL /nologo /D "NDEBUG" /win32
MTL_PROJ=/nologo /D "NDEBUG" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
BSC32_FLAGS=/nologo /o"$(OUTDIR)/Line21.bsc"
BSC32_SBRS=
LINK32=link.exe
```

```
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo
/subsystem:windows /dll /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib
shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo
/subsystem:windows /dll /machine:I386
LINK32_FLAGS=kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib\
 advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib\
 odbccp32.lib /nologo /subsystem:windows /dll /incremental:no\
 /pdb:"$(OUTDIR)/Line21.pdb" /machine:I386 /def:".\L21Dec.def"\
 /out:"$(OUTDIR)/Line21.dll" /implib:"$(OUTDIR)/Line21.lib"
DEF_FILE= \
	".\L21Dec.def"
LINK32_OBJS= \
	"$(INTDIR)/XDSROM.obj" \
	"$(INTDIR)/stable.obj" \
	"$(INTDIR)/L21Dec.obj" \
	"$(INTDIR)/XDSMessages.obj" \
	"$(INTDIR)/Tests.obj" \
	"$(INTDIR)/funct.obj" \
	"$(INTDIR)/Parity.obj"

"$(OUTDIR)\Line21.dll" : "$(OUTDIR)" $(DEF_FILE) $(LINK32_OBJS)
   $(LINK32) @<<
  $(LINK32_FLAGS) $(LINK32_OBJS)
<<

!ELSEIF  "$(CFG)" == "Line21 - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir "Debug"
# PROP Intermediate_Dir "Debug"
# PROP Target_Dir ""
OUTDIR=.\Debug
INTDIR=.\Debug

ALL : "$(OUTDIR)\Line21.dll"

CLEAN :
	-@erase ".\Debug\vc40.pdb"
```

```
        -@erase ".\Debug\vc40.idb"
        -@erase ".\Debug\Line21.dll"
        -@erase ".\Debug\stable.obj"
        -@erase ".\Debug\L21Dec.obj"
        -@erase ".\Debug\XDSMessages.obj"
        -@erase ".\Debug\Tests.obj"
        -@erase ".\Debug\Parity.obj"
        -@erase ".\Debug\funct.obj"
        -@erase ".\Debug\XDSROM.obj"
        -@erase ".\Debug\Line21.ilk"
        -@erase ".\Debug\Line21.lib"
        -@erase ".\Debug\Line21.exp"
        -@erase ".\Debug\Line21.pdb"

"$(OUTDIR)" :
    if not exist "$(OUTDIR)/$(NULL)" mkdir "$(OUTDIR)"

# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D
"_WINDOWS" /YX /c
# ADD CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D
"_WINDOWS" /YX /c
CPP_PROJ=/nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D
"_WINDOWS"\
 /Fp"$(INTDIR)/Line21.pch" /YX /Fo"$(INTDIR)/" /Fd"$(INTDIR)/" /c
CPP_OBJS=.\Debug/
CPP_SBRS=
# ADD BASE MTL /nologo /D "_DEBUG" /win32
# ADD MTL /nologo /D "_DEBUG" /win32
MTL_PROJ=/nologo /D "_DEBUG" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
BSC32_FLAGS=/nologo /o"$(OUTDIR)/Line21.bsc"
BSC32_SBRS=
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo
/subsystem:windows /dll /debug /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib
shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo
/subsystem:windows /dll /debug /machine:I386
LINK32_FLAGS=kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib\
 advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib\
 odbccp32.lib /nologo /subsystem:windows /dll /incremental:yes\
/pdb:"$(OUTDIR)/Line21.pdb" /debug /machine:I386 /def:".\L21Dec.def"\
/out:"$(OUTDIR)/Line21.dll" /implib:"$(OUTDIR)/Line21.lib"
DEF_FILE= \
        ".\L21Dec.def"
LINK32_OBJS= \
        "$(INTDIR)/stable.obj" \
        "$(INTDIR)/L21Dec.obj" \
        "$(INTDIR)/XDSMessages.obj" \
        "$(INTDIR)/Tests.obj" \
        "$(INTDIR)/Parity.obj" \
        "$(INTDIR)/funct.obj" \
        "$(INTDIR)/XDSROM.obj"

"$(OUTDIR)\Line21.dll" : "$(OUTDIR)" $(DEF_FILE) $(LINK32_OBJS)
  $(LINK32) @<<
 $(LINK32_FLAGS) $(LINK32_OBJS)
<<

!ENDIF

.c{$(CPP_OBJS)}.obj:
  $(CPP) $(CPP_PROJ) $<

.cpp{$(CPP_OBJS)}.obj:
  $(CPP) $(CPP_PROJ) $<

.cxx{$(CPP_OBJS)}.obj:
  $(CPP) $(CPP_PROJ) $<

.c{$(CPP_SBRS)}.sbr:
  $(CPP) $(CPP_PROJ) $<

.cpp{$(CPP_SBRS)}.sbr:
  $(CPP) $(CPP_PROJ) $<

.cxx{$(CPP_SBRS)}.sbr:
  $(CPP) $(CPP_PROJ) $<

################################################################################
#####
# Begin Target

# Name "Line21 - Win32 Release"
# Name "Line21 - Win32 Debug"

!IF  "$(CFG)" == "Line21 - Win32 Release"
```

```
!ELSEIF  "$(CFG)" == "Line21 - Win32 Debug"

!ENDIF

################################################################################
#####
# Begin Source File

SOURCE=.\Tests.c
DEP_CPP_TESTS=\
	".\Smachine.h"\
	".\L21Dec.h"\


"$(INTDIR)\Tests.obj" : $(SOURCE) $(DEP_CPP_TESTS) "$(INTDIR)"


# End Source File
################################################################################
#####
# Begin Source File

SOURCE=.\L21Dec.c
DEP_CPP_L21DE=\
	".\L21Dec.h"\
	".\Smachine.h"\


"$(INTDIR)\L21Dec.obj" : $(SOURCE) $(DEP_CPP_L21DE) "$(INTDIR)"


# End Source File
################################################################################
#####
# Begin Source File

SOURCE=.\Parity.c

"$(INTDIR)\Parity.obj" : $(SOURCE) "$(INTDIR)"


# End Source File
################################################################################
#####
# Begin Source File
```

```
SOURCE=.\stable.c
DEP_CPP_STABL=\
        ".\Smachine.h"\
        ".\L21Dec.h"\


"$(INTDIR)\stable.obj" : $(SOURCE) $(DEP_CPP_STABL) "$(INTDIR)"


# End Source File
###########################################################################
#####
# Begin Source File

SOURCE=.\funct.c
DEP_CPP_FUNCT=\
        ".\L21Dec.h"\
        ".\Smachine.h"\


"$(INTDIR)\funct.obj" : $(SOURCE) $(DEP_CPP_FUNCT) "$(INTDIR)"


# End Source File
###########################################################################
#####
# Begin Source File

SOURCE=.\L21Dec.def

!IF  "$(CFG)" == "Line21 - Win32 Release"

!ELSEIF  "$(CFG)" == "Line21 - Win32 Debug"

!ENDIF

# End Source File
###########################################################################
#####
# Begin Source File

SOURCE=.\XDSROM.c
DEP_CPP_XDSRO=\
        ".\L21Dec.h"\
        ".\Smachine.h"\
"$(INTDIR)\XDSROM.obj" : $(SOURCE) $(DEP_CPP_XDSRO) "$(INTDIR)"


# End Source File
###############################################################################
#####
# Begin Source File

SOURCE=.\XDSMessages.c
DEP_CPP_XDSME=\
	".\L21Dec.h"\
	".\Smachine.h"\


"$(INTDIR)\XDSMessages.obj" : $(SOURCE) $(DEP_CPP_XDSME) "$(INTDIR)"


# End Source File
# End Target
# End Project
###############################################################################
#####
```

09651505 .050904

File: Line21.mdp

```
JSG0__E_ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿàÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ:,D"_ÿÿÿÿ_ÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿàÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ___vcbks40.mvb0_(_15 Œçs<mE& Œçs0¤m____
_
        ___b0_(_15 Œçs<mE& Œçs0¤m________
        __Ë]___ÿÿÿÿ__Watch1_Watch2_Watch3_Watch4MLJL15 Œçs<mE& Œçs0¤m____
___!
        AœË]___ÿÿÿÿ__Watch1_Watch2_Watch3_Watch4MLJL15 Œçs<mE& Œçs0¤m#$%
&'()*+ Aœ__LINE21 - WIN32 DEBUG__{/nologo /MTd /W3 /Gm /GX /Zi /Od /D
"WIN32" /D "_DEBUG" /D "_WINDOWS" /Fp"Debug/Line21.pch" /YX /Fo"Debug/"
/Fd"Debug/" /c __/nologo /D "_DEBUG" /win32 __/l 0x409 /d "_DEBUG" __/nologo
/o"Debug/Line21.bsc" _ÿ2_kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo
/subsystem:windows /dll /incremental:yes /pdb:"Debug/Line21.pdb" /debug /machine:I386
/def:".\L21Dec.def" /out:"Debug/Line21.dll" /implib:"Debug/Line21.lib"
_J:\Vb60\Line21\Tests.c___J:\Vb60\Line21\L21Dec.c___J:\Vb60\Line21\Parity.c___J:\Vb6
0\Line21\stable.c___J:\Vb60\Line21\funct.c___J:\Vb60\Line21\L21Dec.def_J:\Vb60\Line21\
XDSROM.c___J:\Vb60\Line21\XDSMessages.c___LINE21 - WIN32 RELEASE__j/nologo
```

```
/MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /Fp"Release/Line21.pch"
/YX /Fo"Release/" /c __/nologo /D "NDEBUG" /win32 __/l 0x409 /d "NDEBUG" __/nologo
/o"Release/Line21.bsc" _ÿ0_kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo
/subsystem:windows /dll /incremental:no /pdb:"Release/Line21.pdb" /machine:I386
/def:".\L21Dec.def" /out:"Release/Line21.dll" /implib:"Release/Line21.lib"
_J:\Vb60\Line21\Tests.c___J:\Vb60\Line21\L21Dec.c___J:\Vb60\Line21\Parity.c___J:\Vb6
0\Line21\stable.c___J:\Vb60\Line21\funct.c___J:\Vb60\Line21\L21Dec.def_J:\Vb60\Line21\
XDSROM.c___J:\Vb60\Line21\XDSMessages.c___Line21 - Win32 Release
Line21.mak__CProject_Line21 - Win32 Release_Line21 - Win32 Release_ê___Line21 -
Win32 Debug_ê_____SSBR__CTargetItem___SSBR       .\Tests.c_
        CFileItem___SSBR
.\L21Dec.c_   CFileItem___SSBR
.\Parity.c_   CFileItem___SSBR
.\stable.c_   CFileItem___SSBR   .\funct.c_      CFileItem___SSBR.\L21Dec.def_
        CFileItem___SSBR
.\XDSROM.c_         CFileItem___SSBR_.\XDSMessages.c_
        CFileItem___SSBR_dep__CDependencyContainer___SSBR
Smachine.h__CDependencyFile___SSBR_L21Dec.h__CDependencyFile___SSBR___DJW_
dep__CDependencyContainer___SSBR
Smachine.h__CDependencyFile___SSBR_L21Dec.h__CDependencyFile___SSBR___DJW_
__DJW___DJWE& Œçst¥m-./0123456789
__m_LINE21 - WIN32 DEBUG__{/nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D
"_DEBUG" /D "_WINDOWS" /Fp"Debug/Line21.pch" /YX /Fo"Debug/" /Fd"Debug/" /c
__/nologo /D "_DEBUG" /win32 __/l 0x409 /d "_DEBUG" __/nologo /o"Debug/Line21.bsc"
_ÿ2_kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib
ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll
/incremental:yes /pdb:"Debug/Line21.pdb" /debug /machine:I386 /def:".\L21Dec.def"
/out:"Debug/Line21.dll" /implib:"Debug/Line21.lib"
_J:\Vb60\Line21\Tests.c___J:\Vb60\Line21\L21Dec.c___J:\Vb60\Line21\Parity.c___J:\Vb6
0\Line21\stable.c___J:\Vb60\Line21\funct.c___J:\Vb60\Line21\L21Dec.def_J:\Vb60\Line21\
XDSROM.c___J:\Vb60\Line21\XDSMessages.c___LINE21 - WIN32 RELEASE__j/nologo
/MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /Fp"Release/Line21.pch"
/YX /Fo"Release/" /c __/nologo /D "NDEBUG" /win32 __/l 0x409 /d "NDEBUG" __/nologo
/o"Release/Line21.bsc" _ÿ0_kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo
/subsystem:windows /dll /incremental:no /pdb:"Release/Line21.pdb" /machine:I386
/def:".\L21Dec.def" /out:"Release/Line21.dll" /iE& Œçst¥m;<=>?@ABC
```

File: Line21.ncb

```
Microsoft C/C++ program database 2.00
_JG__)À¤üž(þÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

09654505.050501

ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ

ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿòÿÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ

09851605.050901

ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ

ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ
ÿÿÿ|__□o__e__Y_s__ –____3
d__‰_‡___K__o_ã___Ú_
#□__|(5__U-A__b2□*__H7□___;<,__°Az__£Fú_"K___*Pí__U"—¢³À"—¢³À"—¢³À"—¢³À"—¢³À"—¢³À"—¢³À"—¢³À"—¢³À"—¢³À"—¢³À"—¢³À"—¢³À"—¢³À"—¢³À"—¢³À_l0_{l™9___,ðž$_öž,öž<öž__±__@_
__G
§__ø1
¬__3

ú__@ D
_□ß__@ C
_Î__@ B
_□º__Ñ@ A
_2__ý@ I
____@ H
_B__@ J
_ä__@ Q
_O__@ K
□{__@ L

¤__@ M
_Á__@ N
Ò__@ O

_ø________ ___
__
________"î_"É_(_(_(_ö__Ž_±_""s__(_Ï___
□æ________d__)_□þ____Š___!□‡__&□Ý___+K__%0_ã__5____9Ú__>__'C□____H5_
_$Mò___RA__#W____*__"an___d□P___g"___l□___&q¢__ t-
___y,__+~z__*ƒž___†¦___‹,__
□ú_ •À___š□È___Ÿ___!¤¯__©¯___®í__³x___"—¢³Ž_"—¢³Ž_"—¢³Ž_"—¢³Ž_"—
¢³Ž_"—¢³Ž_…_³Ž_"—¢³Ž_"—¢³Ž_"—¢³Ž_"—¢³Ž_'_—¢B_"—¢³Ž_"—¢³Ž_"—¢³Ž_"—
¢³Ž_"—¢³Ž_"—¢³Ž_"—¢³Ž_"—¢³Ž_'_[_i_'_[_i_"—¢³Ž_"—¢³Ž_…_³Ž_"—¢³Ž_"—
¢³Ž_"—¢³Ž_…_³Ž_"—¢³Ž_"—¢³Ž_"—¢³Ž_"—¢³Ž_"—¢³Ž_"—¢³Ž_"—¢³Ž_"—¢³Ž_"—
¢³Ž_…_—¢³|_ _□o_ _[__ _O__ □'__@ __e_____ž±__Ð

____ _
□n__Y
P__T_□§__ _x_____x__2_™_o_¹_Ç_Ô_ñ_'_—¢³'_____'_____Ž_□_…_—¢³x_____:_|_
_o_ _□e__Y_Ï__<_Ï__;_æ__X_æ__$_æ__3_æ__l_□h__ü___Ñ__n žÑ__%_Ñ__E
Ñ__M_Ñ__=_Ñ__`_Ñ__Y_□þ__%_þ__m_þ__n_e_____žÝ__:_Ý__m_Ý__
;_žÁ__$_Á__h Á__g"Á___$žÁ__"&Á__ü!(Á__.*žÁ__i,□Á__l.Á__#0Á__U2žÁ__04Á__
6Á__L8□Á__1:Á__k<žÁ__2>Á__D@Á__jBÁ__3DÁ__XFžÁ__WHÁ__4J□Á__<L□Á__
mNÁ__5PÁ__;RÁ__/TÁ__6VÁ__VXÁ__7Z□Á__:\Á__9^Á__8`___ 6aò__ 4bò__ 9c___
5dž¶__Tf¶___h□¶__-
j□¶___l¶__fn¶__Kp□¶__Cr¼__;t¼__7v¼__8x¼__<z□¼__D|¼__3~¼__6€¼__:,□¼__2„ž
¼__4†¼__Lˆ¼__5Š¼__/Œ¼__1Ž¼__.□¼__U'¼__"¼__V–
¼__k˜¼__0š¼__Wœ□¼__!ž¼__9 ¼__X¢¼__h¤ž¼__j¦¼__i¨□¼__$ª¼__m¬□¼__l®¼__#
°ž¼__"²¼__g´¼___¶"__ 8 ž©____¸-__ 7¹¦__ø !º¦__ i»¦__ 1¼□¦__j½¦__ W¾¦__ V¿□¦__
0À¦__"Á□¸__ MÂƒ_____ÃÀ__ YÄ□___ ÅÈ__`Æ□□___
Ç¯__
EÈžu_____ÉÆ_Æ_Æ_Æ_Æ_Æ_o_×_©_×_e_×_□_×_□_×_e_×_e_×_□_È_Æ_È_o_Æ_Æ_
Æ_"h_"©_"©_"e_"ƒ_"u_"h_"u_"h_"□_"□_"u_"e_"ƒ_"ƒ_"□_"□_"u_"ƒ_"h_"h_"ƒ_"e_"e_
"©_"e_"e_"e_"e_"u_"©_"e_"e_"e_Æ_Æ_Æ_Æ_"□_"e_"h_"□_"©_"ƒ_"u_"s_"z_"__"d_"
Ú"__"5_"_"__"A_"Ú"*_"ã"ú"Ú"__"ã"í"__"í"ú"í"K_"__"ã"ú"í"__",_"__"__"ú"Ú"ÚÆ_o_
Æ_Æ_Æ_Æ_Æ_Æ_Æ_Æ_Æ_È_o_È_o_È_o_È_o_|_ _o_ _□e__ _ö__ _Y_ —___

```
_Ï__ó_æ___m_□þ__@ w_Ý__ ___§__Â_______+_____□___¯!ò__€ˆ&___›+"__Ø0-
___é5□¦___:,__+?À__ >D□g___ EÈ__ MJ5____K¯ __P□_"—¢³À"—¢³À"—¢³À"—
¢³À"—¢³À____"—¢³À"—¢³À"—¢³À"—¢³À"—¢³À"—¢³À"—¢³À"—¢³Ào_"—¢³À<_"—
¢³À|__ _o___□Y__ —_ü 6_______ __+___ >_□g___ ê_5__ü x_ê__ü &__□Ò__ $__□Þ__
%__ü__ü (_  □_____o_<_ó_□_□_□_|__o___že__ _ö___□Y__ □Š__ü _¢__ü = ž__ü _
        □5__ü__
ê___  _Ò___Þ____
ü__ü_
_□…_³À…_³À…_³À<_ó_□_□_□_________  ________-_  B,__U_€_€___@__
_______¨×»8□_ä___

€___@€____€_____‚Ð°8÷ž__-_'·_B _A _!@_"__€_€Š_'@_,__Bæè°87_±_
_€@_€_€
_____ðÐ°8□"_x_____´./P_Ñ_{É_„@ˆ_'@_,__B____A_H__°?™9ß_§__P_„@ˆ_'@_,__B__
____°_¼8°_ü__€@____, @_@™9__ü_
__„@ €___, @_ °8__LINE21 - WIN32 DEBUGLINE21 - WIN32 RELEASELine21 -
Win32
DebugJ:\Vb60\Line21\Tests.c<windows.h><stdio.h>"sMachine.h""L21Dec.h"DefaultTestint
long fieldunsigned char chchar mes [ ]struct cState *
cStatePtrNullTestF1XDSTestXDSStartTestXDSContTestXDSEndTestControlTestTextMode
TestRollUpTestPaintOnTestPopOnTestEndOfCaptionTestPACTestCtrlChangeTestCRTestU
RLTestURLEndTestJ:\Vb60\Line21\Smachine.hSPTRSVALXDSStateXDSState::OpenBOO
LXDSState::LengthXDSState::Datachar [ 33
]XDSState::CheckSumcStatecState::CurrentCtrlint [ 2
]cState::ChannelcState::ModecState::Textchar [ 2 ] [ 2 ] [ 2 ] [ 256 ]cState::TextLengthint [ 2
] [ 2 ] [ 2 ]cState::Typeint [ 2 ] [ 2
]cState::XDSClasscState::XDSRecordcState::NextStateSVAL* [ 2
]J:\Vb60\Line21\L21Dec.hLine21Decode__declspec() LONG APIENTRYchar mess [
]ShiftRightunsigned longlongShiftLeftstateMachinevoidDefstruct cState
*VcpTypXDSBeginXDSStartXDSContXDSDataXDSEndChannelChangeDumpTextControl
SavePopOnDecodeDataSaveRollUpDecodePaintOnDecodeTextDecodeURLDecodeJ:\Vb60\
Line21\L21Dec.c"windows.h""stdio.h"DataStateSVAL [ ]parityunsigned char [
]CCstatestruct cStatecStatePtrDllMainBOOL APIENTRYHANDLE hInstDWORD
ul_reason_being_calledLPVOID lpReservedunsigned long valuelong
shiftJ:\Vb60\Line21\Parity.cunsigned char [ 256
]J:\Vb60\Line21\stable.cControlStateXDSStartStateXDSContStateXDSDataStateXDSEndSta
teURLDataStateSTATETESTNEXT&DEFAULT&DNEXTDEFAULTJ:\Vb60\Line21\funct
.c<string.h>MaxLengthchar [ CLASSMAX ] [ RECORDMAX ]MinLengthXDSMeschar [
CLASSMAX ] [ RECORDMAX ] [ MESLEN ]XDSDispvoid(* [ CLASSMAX ] [
RECORDMAX ] ) ( )CTypechar* [
]EndOfCaptionDecodeJ:\Vb60\Line21\XDSROM.cmpaaRatingstvpgRatingsflagmaskchar [
]typesJ:\Vb60\Line21\XDSMessages.cÑ__o_ž_*_Ñ_5_¶_È_É___—
_³ò___ƒ_È_ó_▒_æ_[_e°_u_
_±___þ_z_d_A_¼_'___±_s_ö_n_g_□_§_h__2_úÞ_¢B_,_\_+_O_<_YP_ä_i_B…_e_ü_¤_íÚ
±_×_"_§_À_ß_Ò___o___©___™_Á_7_O_Ž_Ô_—
{_Ý_À__Á_°_Š_oö_:_x_Ž_'___Ò_ú_"§_□_ñ_(_x_P_¹___ê_[_Ï_|î_B_‡□_5_ãÇ___¦_Æ_Î_
```

K_,__-ß__¬_-
_ ___¢_"_□_‡_l0_{l™9′_/names/ncb/targetinfo/ncb/moduleinfo/ncb/module/J:\Vb60\Line21\Tests.c/ncb/module/J:\Vb60\Line21\Smachine.h/ncb/module/J:\Vb60\Line21\L21Dec.h/ncb/module/J:\Vb60\Line21\L21Dec.c/ncb/module/J:\Vb60\Line21\Parity.c/ncb/module/J:\Vb60\Line21\stable.c/ncb/module/J:\Vb60\Line21\funct.c/ncb/module/J:\Vb60\Line21\XDSROM.c/ncb/module/J:\Vb60\Line21\XDSMessages.c/ncb/target/Line21 - Win32 Debug/ncb/target/LINE21 - WIN32 RELEASE___,_?_'_"
_____'__p__,_G__p #___
ÜJ__(,õžL__öž,öž<öžo
Èÿž_¨_□_üõž¤_„÷žx_0ûžL_8÷ž(_,øž_ÈúžÔ
öžÐ_,_□äÈ_□__4üž_¨ÿž_Øÿž_'$%&!"#____________

File: Line21Def.bas

```
Attribute VB_Name = "Line21Def"
Public Const APPNAME As String = "Line21Watch"
Public Const SETTING As String = "Settings"
Public Const TCOMF As String = "COMF"
Public Const IP As String = "IP"
Public Const PORT As String = "Port"

Option Explicit

Declare Function Line21Decode Lib "Line21.dll" _
  (ByVal field As Long, ByVal ch As Byte, ByVal mes As String) _
  As Long

Declare Function ShiftRight Lib "Line21.dll" _
  (ByVal Val As Long, ByVal Shift As Long) As Long

Declare Function ShiftLeft Lib "Line21.dll" _
  (ByVal Val As Long, ByVal Shift As Long) As Long
```

File: Line21Watch.vbp

```
Type=Exe
Reference=*\G{00020430-0000-0000-C000-
000000000046}#2.0#0#C:\WINDOWS\SYSTEM\StdOle2.tlb#OLE Automation
Object={648A5603-2C6E-101B-82B6-000000000014}#1.1#0; MSCOMM32.OCX
Object={248DD890-BB45-11CF-9ABC-0080C7E7B78D}#1.0#0; MSWINSCK.OCX
Form=FrmLine21Watch.frm
Module=Line21Def; Line21Def.bas
Startup="FrmLine21Watch"
HelpFile=""
Command32=""
Name="Line21Watch"
HelpContextID="0"
CompatibleMode="0"
MajorVer=1
MinorVer=0
RevisionVer=0
AutoIncrementVer=0
ServerSupportFiles=0
VersionCompanyName="norpak"
CompilationType=0
OptimizationType=0
FavorPentiumPro(tm)=0
CodeViewDebugInfo=0
NoAliasing=0
BoundsCheck=0
OverflowCheck=0
FlPointCheck=0
FDIVCheck=0
UnroundedFP=0
StartMode=0
Unattended=0
Retained=0
ThreadPerObject=0
MaxNumberOfThreads=1
```

File: Line21Watch.vbw

```
FrmLine21Watch = 115, 120, 649, 565, C, 46, 48, 580, 493, C
Line21Def = 69, 72, 603, 517, C
```

File: Parity.c

Copyright © 2000 I LOVE TV INC. (868970 Ontario Inc.)
All rights reserved

```
unsigned char parity[256] = {
 0x80, 0x01, 0x02, 0x83, 0x04, 0x85, 0x86, 0x07,
 0x08, 0x89, 0x8a, 0x0b, 0x8c, 0x0d, 0x0e, 0x8f,
 0x10, 0x91, 0x92, 0x13, 0x94, 0x15, 0x16, 0x97,
 0x98, 0x19, 0x1a, 0x9b, 0x1c, 0x9d, 0x9e, 0x1f,
 0x20, 0xa1, 0xa2, 0x23, 0xa4, 0x25, 0x26, 0xa7,
 0xa8, 0x29, 0x2a, 0xab, 0x2c, 0xad, 0xae, 0x2f,
 0xb0, 0x31, 0x32, 0xb3, 0x34, 0xb5, 0xb6, 0x37,
 0x38, 0xb9, 0xba, 0x3b, 0xbc, 0x3d, 0x3e, 0xbf,
 0x40, 0xc1, 0xc2, 0x43, 0xc4, 0x45, 0x46, 0xc7,
 0xc8, 0x49, 0x4a, 0xcb, 0x4c, 0xcd, 0xce, 0x4f,
 0xd0, 0x51, 0x52, 0xd3, 0x54, 0xd5, 0xd6, 0x57,
 0x58, 0xd9, 0xda, 0x5b, 0xdc, 0x5d, 0x5e, 0xdf,
 0xe0, 0x61, 0x62, 0xe3, 0x64, 0xe5, 0xe6, 0x67,
 0x68, 0xe9, 0xea, 0x6b, 0xec, 0x6d, 0x6e, 0xef,
 0x70, 0xf1, 0xf2, 0x73, 0xf4, 0x75, 0x76, 0xf7,
 0xf8, 0x79, 0x7a, 0xfb, 0x7c, 0xfd, 0xfe, 0x7f,
 0x80, 0x01, 0x02, 0x83, 0x04, 0x85, 0x86, 0x07,
 0x08, 0x89, 0x8a, 0x0b, 0x8c, 0x0d, 0x0e, 0x8f,
 0x10, 0x91, 0x92, 0x13, 0x94, 0x15, 0x16, 0x97,
 0x98, 0x19, 0x1a, 0x9b, 0x1c, 0x9d, 0x9e, 0x1f,
 0x20, 0xa1, 0xa2, 0x23, 0xa4, 0x25, 0x26, 0xa7,
 0xa8, 0x29, 0x2a, 0xab, 0x2c, 0xad, 0xae, 0x2f,
 0xb0, 0x31, 0x32, 0xb3, 0x34, 0xb5, 0xb6, 0x37,
 0x38, 0xb9, 0xba, 0x3b, 0xbc, 0x3d, 0x3e, 0xbf,
 0x40, 0xc1, 0xc2, 0x43, 0xc4, 0x45, 0x46, 0xc7,
 0xc8, 0x49, 0x4a, 0xcb, 0x4c, 0xcd, 0xce, 0x4f,
 0xd0, 0x51, 0x52, 0xd3, 0x54, 0xd5, 0xd6, 0x57,
 0x58, 0xd9, 0xda, 0x5b, 0xdc, 0x5d, 0x5e, 0xdf,
 0xe0, 0x61, 0x62, 0xe3, 0x64, 0xe5, 0xe6, 0x67,
 0x68, 0xe9, 0xea, 0x6b, 0xec, 0x6d, 0x6e, 0xef,
 0x70, 0xf1, 0xf2, 0x73, 0xf4, 0x75, 0x76, 0xf7,
 0xf8, 0x79, 0x7a, 0xfb, 0x7c, 0xfd, 0xfe, 0x7f};
```

File: Smachine.h

```
#define WINKCLASS 0x42
#define WINKSUMSTART  WINKCLASS + 0x0D
#define XDSEND 0x0F

#define OVERWRITE   " Error:  Overwrite"
#define CONTNOSTART " Error:  Continue with no Start"
#define TOOLONG     " Error:  too many bytes"
#define BADCHECKSUM " Error:  bad check sum"
#define TOOSHORT    " Error:  too few bytes"

#define XDSSTART "XDS "
#define C1START  "C1  "
#define C2START  "C2  "
#define T1START  "T1  "
#define T2START  "T2  "
#define C3START  "C3  "
#define C4START  "C4  "
#define T3START  "T3  "
#define T4START  "T4  "
#define ONESP    " "
#define TWOSP    "  "
#define SEVENSP  "       "
#define MPAA     " MPAA - "
#define TVPG     " TVPG - "
#define OTHER    " OTHER - non-USA"
#define TVPG_FV  ",FV"
#define TVPG_V   ",V"
#define TVPG_S   ",S"
#define TVPG_L   ",L"
#define TVPG_D   ",D"

#define TRUE 1
#define FALSE 0

#define C1 0
#define C2 1
#define CAP 0
#define TXT 1
#define FIELD1 1
#define FIELD2 2
#define POPUP 0
#define PAINTON 1
#define ROLLUP2 2
```

```
#define ROLLUP3 3
#define ROLLUP4 4
#define T1 1
#define T2 2


typedef int (__cdecl *FPTR)(), *SPTR;

typedef union sval{
 FPTR fptr;
 SPTR dptr;
 }SVAL;

#define CLASSMAX 7
#define RECORDMAX 128
#define MESLEN 6

#define XDSWink 0
#define XDSVChip 1
#define XDSMax 2

struct XDSState {
 BOOL Open;
 int Length;
 char Data[33];
 int CheckSum;
 };

struct cState {
 int CurrentCtrl[2];                // Save control character which starts pair
 int Channel[2];                   // Channel 1 or 2 (control high or low)
 int Mode[2];                     // Mode captioning or text
 char Text[2][2][2][256];            // The saved text
 int TextLength[2][2][2];            // The length of the saved text
 int Type[2][2];                   // The type of caption
 int XDSClass;                     // The current XDS class
 int XDSRecord;                     // The current XDS record
 struct XDSState XDS[CLASSMAX][RECORDMAX]; // stored data for current XDS state
 SVAL *NextState[2];                  // pointer to the next state
 };

#define STATE(x) SVAL x##[] = {
#define TEST(x) &##x##,
#define NEXT(x)  NULL, ##x##,
#define DEFAULT &DefaultTest,
#define DNEXT(x) NULL, ##x##};
```

File: stable.c

```
#include <windows.h>
#include <stdio.h>
#include "sMachine.h"
#include "L21Dec.h"

extern SVAL DataState[];
extern SVAL ControlState[];
extern SVAL XDSStartState[];
extern SVAL XDSContState[];
extern SVAL XDSDataState[];
extern SVAL XDSEndState[];
extern SVAL URLDataState[];

#pragma warning(disable : 4047)
/*
 * State Tables
 *
 * This is what drives the state machine. Each state has one or more possible
 * sub-state changes. Each of these is made up of a test followed by a list
 * of functions to execute if the test passes and a new state to go to. The
 * state machine tries each test in order. The first test that passes has its
 * functions executed and the state changed to its next state. The state
 * machine then exits without trying any more states. The last test on the list
 * for a state must be the default test which always returns true.
 */

// State DataState
// This state is to go when we have text data
//

STATE(DataState)
  TEST(NullTest) NEXT(DataState)
  TEST(F1XDSTest) NEXT(DataState)
  TEST(XDSStartTest) &XDSBegin, NEXT(XDSStartState)
  TEST(XDSContTest) &XDSBegin, NEXT(XDSContState)
  TEST(XDSEndTest) NEXT(XDSEndState)
  TEST(ControlTest) &ControlSave, NEXT(ControlState)
  DEFAULT &DataSave, DNEXT(DataState)

// State ControlState
// In this state we have received a control character and the next
// character will decide what action we should take. Warning the control
// pair may be repeated but in this case we do not care. Normally you
// require extra states to filter out repeats.

STATE(ControlState)
  TEST(NullTest) NEXT(ControlState)
  TEST(F1XDSTest) NEXT(DataState)
  TEST(XDSStartTest) &XDSBegin, NEXT(XDSStartState)
  TEST(XDSContTest) &XDSBegin, NEXT(XDSContState)
  TEST(XDSEndTest) NEXT(XDSEndState)
  TEST(ControlTest) &ControlSave, NEXT(ControlState)
  TEST(PopOnTest) &PopOnDecode, NEXT(DataState)
  TEST(RollUpTest) &RollUpDecode, NEXT(DataState)
  TEST(PaintOnTest) &PaintOnDecode, NEXT(DataState)
  TEST(URLTest) &URLDecode, NEXT(URLDataState)
  TEST(TextModeTest) &TextDecode, NEXT(DataState)
  TEST(EndOfCaptionTest) &PopOnDecode, NEXT(DataState)
  TEST(PACTest) &DumpText, NEXT(DataState)
  TEST(CRTest) &ChannelChange, &DumpText, NEXT(DataState)
  TEST(CtrlChangeTest) &ChannelChange, NEXT(DataState)
  DEFAULT DNEXT(DataState)

// State XDSStartState
// In this case we have received a an XDS Undefined Start character (0D)
// The next character determines which class function to execute.

STATE(XDSStartState)
  TEST(NullTest) NEXT(XDSStartState)
  DEFAULT &XDSStart, DNEXT(XDSDataState)

// State XDSContState
// In this case we have received a an XDS Undefined Continue character (0E)
// The next character determines which class function to execute.

STATE(XDSContState)
  TEST(NullTest) NEXT(XDSContState)
  DEFAULT &XDSCont, DNEXT(XDSDataState)

// State XDSDataState
// In this state we have received the first character of
// the XDS stream. Want to stay in this state until
// we have receive the full XDS Message.

STATE(XDSDataState)
  TEST(XDSEndTest) NEXT(XDSEndState)
  TEST(XDSStartTest) &XDSBegin, NEXT(XDSStartState)
  TEST(XDSContTest) &XDSBegin, NEXT(XDSContState)
  TEST(ControlTest) &ControlSave, NEXT(ControlState)
```

```
 DEFAULT &XDSData, DNEXT(XDSDataState)

// State XDSEndState
//  We have received the XDS end command (0f) we need to fetch the
//  checksum and then we are done and have recieved the full message.

STATE(XDSEndState)
 DEFAULT &XDSEnd, DNEXT(DataState)

// State URLDataState
//  This state is to go when we have T2 (URL) data
//

STATE(URLDataState)
 TEST(NullTest) NEXT(URLDataState)
 TEST(F1XDSTest) NEXT(URLDataState)
 TEST(XDSStartTest) &XDSBegin, NEXT(XDSStartState)
 TEST(XDSContTest) &XDSBegin, NEXT(XDSContState)
 TEST(XDSEndTest) NEXT(XDSEndState)
 TEST(ControlTest) &ControlSave, NEXT(ControlState)
 TEST(URLEndTest) &DataSave, &DumpText, NEXT(URLDataState)
 DEFAULT &DataSave, DNEXT(URLDataState)
```

File: Tests.c

```
#include <windows.h>
#include <stdio.h>
#include "sMachine.h"
#include "L21Dec.h"

int DefaultTest(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
  {
  return(TRUE);
  }

int NullTest(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
  {
  if(ch == 0x00)
   return(TRUE);
  else
   return(FALSE);
  }

int F1XDSTest(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
  {
  if((ch < 0x10) && (field == FIELD1))
   return(TRUE);
  else
   return(FALSE);
  }

int XDSStartTest(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
  {
  if((ch > 0) && (ch < 0x0f) && ((ch & 1) == 1))
   return(TRUE);
  else
   return(FALSE);
  }

int XDSContTest(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
  {
  if((ch > 0) && (ch < 0x0f) && ((ch & 1) == 0))
   return(TRUE);
  else
   return(FALSE);
  }
```

```
int XDSEndTest(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
 {
 if(ch == 0x0f)
  return(TRUE);
 else
  return(FALSE);
 }


int ControlTest(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
 {
 if((ch > 0x0f) && (ch < 0x20))
  return(TRUE);
 else
  return(FALSE);
 }

int TextModeTest(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
 {
 int c;

 c = cStatePtr->CurrentCtrl[field - 1];
 if((c == 0x14) || (c == 0x15) || (c == 0x1c) || (c == 0x1d))
  {
  if((ch == 0x2a) || (ch == 0x2b))
   return(TRUE);
  }
 return(FALSE);
 }

int RollUpTest(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
 {
 int c;

 c = cStatePtr->CurrentCtrl[field - 1];
 if((c == 0x14) || (c == 0x15) || (c == 0x1c) || (c == 0x1d))
  {
  if((ch > 0x24) && (ch < 0x28))
   return(TRUE);
  }
 return(FALSE);
 }

int PaintOnTest(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
 {
 int c;
```

```
  c = cStatePtr->CurrentCtrl[field - 1];
  if((c == 0x14) || (c == 0x15) || (c == 0x1c) || (c == 0x1d))
    {
    if(ch == 0x29)
      return(TRUE);
    }
  return(FALSE);
  }

int PopOnTest(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
  {
  int c;

  c = cStatePtr->CurrentCtrl[field - 1];
  if((c == 0x14) || (c == 0x15) || (c == 0x1c) || (c == 0x1d))
    {
    if(ch == 0x20)
      return(TRUE);
    }
  return(FALSE);
  }

int EndOfCaptionTest(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
  {
  int c;

  c = cStatePtr->CurrentCtrl[field - 1];
  if((c == 0x14) || (c == 0x15) || (c == 0x1c) || (c == 0x1d))
    {
    if(ch == 0x2f)
      return(TRUE);
    }
  return(FALSE);
  }

int PACTest(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
  {
  int c;

  c = cStatePtr->CurrentCtrl[field - 1];
  if((c > 0xf) && (c < 0x20))
    {
    if(ch > 0x3f)
      return(TRUE);
    }
```

```
 return(FALSE);
 }

int CtrlChangeTest(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
 {
 int f, c, ctrl;

 f = field - 1;
 c = cStatePtr->Channel[f];
 ctrl = cStatePtr->CurrentCtrl[f];
 if(((ch > 0x1f) && (ch < 0x40)) && (((ctrl < 0x18) && (c != C1)) || ((ctrl > 0x17) && (c !=
C2))))
   return(TRUE);
 else
   return(FALSE);
 }

int CRTest(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
 {
 int c;

 c = cStatePtr->CurrentCtrl[field - 1];
 if((c == 0x14) || (c == 0x15) || (c == 0x1c) || (c == 0x1d))
   {
   if(ch == 0x2d)
     return(TRUE);
   }
 return(FALSE);
 }

int URLTest(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
 {
 int c;

 c = cStatePtr->CurrentCtrl[field - 1];
 if((field == FIELD1) && (c == 0x1c))
   {
   if((ch == 0x2a) || (ch == 0x2b))
     return(TRUE);
   }
 return(FALSE);
 }

int URLEndTest(long field, unsigned char ch, char mes[], struct cState *cStatePtr)
 {
 int f, l;
```

```
  char *t;

  f = field - 1;
  l = cStatePtr->TextLength[f][C2][TXT];
  t = cStatePtr->Text[f][C2][TXT];
  if(l > 5)
   if((ch == ']') && (t[l-5] == '['))
     {
     if(((t[l-4] >= '0') && (t[l-4] <= '9')) || ((t[l-4] >= 'A') && (t[l-4] <= 'F')) || ((t[l-4] >= 'a')
&& (t[l-4] <= 'f')))
       if(((t[l-3] >= '0') && (t[l-3] <= '9')) || ((t[l-3] >= 'A') && (t[l-3] <= 'F')) || ((t[l-3] >= 'a')
&& (t[l-3] <= 'f')))
        if(((t[l-2] >= '0') && (t[l-2] <= '9')) || ((t[l-2] >= 'A') && (t[l-2] <= 'F')) || ((t[l-2] >= 'a')
&& (t[l-2] <= 'f')))
          if(((t[l-1] >= '0') && (t[l-1] <= '9')) || ((t[l-1] >= 'A') && (t[l-1] <= 'F')) || ((t[l-1] >=
'a') && (t[l-1] <= 'f')))
           return(TRUE);
     }
  return(FALSE);
  }
```

09851605.050901

File: XDSMessages.c

```
#include <windows.h>
#include <stdio.h>
#include <string.h>
#include "L21Dec.h"
#include "sMachine.h"

extern char XDSMes[CLASSMAX][RECORDMAX][MESLEN];
extern char *mpaaRatings[];
extern char *tvpgRatings[];
extern char flagmask[];
extern char *types[];

void Def(char mes[], struct cState *cStatePtr)
 {
 strcpy(mes,XDSSTART);
 strcat(mes,XDSMes[cStatePtr->XDSClass][cStatePtr->XDSRecord]);
 strcat(mes,TWOSP);
 cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].Data[cStatePtr-
>XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].Length] = 0;
 strcat(mes, cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].Data);
 }

void Vcp(char mes[], struct cState *cStatePtr)
 {
 char b1, b2, type;
 char flags;

 strcpy(mes,XDSSTART);
 strcat(mes,XDSMes[cStatePtr->XDSClass][cStatePtr->XDSRecord]);
 b1 = cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].Data[0];
 b2 = cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].Data[1];
 type = ((b1 >> 3) & 3);
 if((type == 0) || (type == 2))  //MPAA
  {
  strcat(mes, MPAA);
  strcat(mes, mpaaRatings[b1 & 7]);
  }
 else if(type == 1) // TVPG
  {
  strcat(mes, TVPG);
  strcat(mes, tvpgRatings[b2 & 7]);
  flags = (b1 >> 5) & 1;
  flags = flags | ((b2 & 0x38) >> 2);
  flags = flags | ((b2 & 0x20) >> 1);
```

```
   flags = flags & flagmask[b2 & 7];
   if(flags & 0x10)
    strcat(mes,TVPG_FV);
   if(flags & 8)
    strcat(mes,TVPG_V);
   if(flags & 4)
    strcat(mes,TVPG_S);
   if(flags & 2)
    strcat(mes,TVPG_L);
   if(flags & 1)
    strcat(mes,TVPG_D);
   }
  else // Other
   {
   strcat(mes, OTHER);
   }
  }

void Typ(char mes[], struct cState *cStatePtr)
  {
  int i;

  strcpy(mes,XDSSTART);
  strcat(mes,XDSMes[cStatePtr->XDSClass][cStatePtr->XDSRecord]);
  strcat(mes,TWOSP);
  for(i = 0;i < cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].Length;i++)
   {
   strcat(mes, types[cStatePtr->XDS[cStatePtr->XDSClass][cStatePtr->XDSRecord].Data[i]
- 0x20]);
   strcat(mes, ONESP);
   }
  }
```

File: XDSROM.c

```
#include <windows.h>
#include "L21Dec.h"
#include "sMachine.h"

char MaxLength[CLASSMAX][RECORDMAX] = {
 { 0,  4,  6, 32, 32,  2,  2,  8, 32,  4, 32, 32, 32, 32, 32, 32, // Current Class
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32},
 { 0,  4,  6, 32, 32,  2,  2,  8,  4, 32, 32, 32, 32, 32, 32, 32,  // Future Class
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32},
 { 0, 32,  6,  2, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,  // Channel Class
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32},
 { 0,  6,  6, 32,  2, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,  // Miscellaneous
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
   2, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32},
 { 0, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,  // Public Service
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32},
{32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,  // Reserved
 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32},
{32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,  // Undefined Class
 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32}};

char MinLength[CLASSMAX][RECORDMAX] = {
 { 0, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, // Current Class
  0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2},
 { 0, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  // Future Class
  0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2},
 { 0, 2, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  // Channel Class
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2},
 { 0, 6, 6, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  // Miscellaneous
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
```

09851605.050901

```
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2},
 { 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, // Public Service
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2},
 { 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, // Reserved
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,},
 { 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, // Undefined Class
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2}};

char XDSMes[CLASSMAX][RECORDMAX][MESLEN] = {
 {"Null ","Start","Durat","Show ","Type ","VChip","Audio","Capti",  // Current Class
  "0108 ","Aspec","010A ","010B ","Comp1","Comp2","010E ","010F ",
  "Desc1","Desc2","Desc3","Desc4","Desc5","Desc6","Desc7","Desc8",
  "0118 ","0119 ","011A ","011B ","011C ","011D ","011E ","011F ",
  "0120 ","0121 ","0122 ","0123 ","0124 ","0125 ","0126 ","0127 ",
  "0128 ","0129 ","012A ","012B ","012C ","012D ","012E ","012F ",
  "0130 ","0131 ","0132 ","0133 ","0134 ","0135 ","0136 ","0137 ",
  "0138 ","0139 ","013A ","013B ","013C ","013D ","013E ","013F ",
  "0140 ","0141 ","0142 ","0143 ","0144 ","0145 ","0146 ","0147 ",
  "0148 ","0149 ","014A ","014B ","014C ","014D ","014E ","014F ",
  "0150 ","0151 ","0152 ","0153 ","0154 ","0155 ","0156 ","0157 ",
  "0158 ","0159 ","015A ","015B ","015C ","015D ","015E ","015F ",
  "0160 ","0161 ","0162 ","0163 ","0164 ","0165 ","0166 ","0167 ",
  "0168 ","0169 ","016A ","016B ","016C ","016D ","016E ","016F ",
"0170 ","0171 ","0172 ","0173 ","0174 ","0175 ","0176 ","0177 ",
"0178 ","0179 ","017A ","017B ","017C ","017D ","017E ","017F "},
{"0300 ","0301 ","0302 ","0303 ","0304 ","0305 ","0306 ","0307 ",  // Future Class
"0308 ","0309 ","030A ","030B ","030C ","030D ","030E ","030F ",
"0310 ","0311 ","0312 ","0313 ","0314 ","0315 ","0316 ","0317 ",
"0318 ","0319 ","031A ","031B ","031C ","031D ","031E ","031F ",
"0320 ","0321 ","0322 ","0323 ","0324 ","0325 ","0326 ","0327 ",
"0328 ","0329 ","032A ","032B ","032C ","032D ","032E ","032F ",
"0330 ","0331 ","0332 ","0333 ","0334 ","0335 ","0336 ","0337 ",
"0338 ","0339 ","033A ","033B ","033C ","033D ","033E ","033F ",
"0340 ","0341 ","0342 ","0343 ","0344 ","0345 ","0346 ","0347 ",
"0348 ","0349 ","034A ","034B ","034C ","034D ","034E ","034F ",
"0350 ","0351 ","0352 ","0353 ","0354 ","0355 ","0356 ","0357 ",
"0358 ","0359 ","035A ","035B ","035C ","035D ","035E ","035F ",
"0360 ","0361 ","0362 ","0363 ","0364 ","0365 ","0366 ","0367 ",
"0368 ","0369 ","036A ","036B ","036C ","036D ","036E ","036F ",
"0370 ","0371 ","0372 ","0373 ","0374 ","0375 ","0376 ","0377 ",
"0378 ","0379 ","037A ","037B ","037C ","037D ","037E ","037F "},
{"0500 ","Netwk","Affil","Delay","0504 ","0505 ","0506 ","0507 ",  // Channel Class
"0508 ","0509 ","050A ","050B ","050C ","050D ","050E ","050F ",
"0510 ","0511 ","0512 ","0513 ","0514 ","0515 ","0516 ","0517 ",
"0518 ","0519 ","051A ","051B ","051C ","051D ","051E ","051F ",
"0520 ","0521 ","0522 ","0523 ","0524 ","0525 ","0526 ","0527 ",
"0528 ","0529 ","052A ","052B ","052C ","052D ","052E ","052F ",
"0530 ","0531 ","0532 ","0533 ","0534 ","0535 ","0536 ","0537 ",
"0538 ","0539 ","053A ","053B ","053C ","053D ","053E ","053F ",
"0540 ","0541 ","0542 ","0543 ","0544 ","0545 ","0546 ","0547 ",
"0548 ","0549 ","054A ","054B ","054C ","054D ","054E ","054F ",
"0550 ","0551 ","0552 ","0553 ","0554 ","0555 ","0556 ","0557 ",
"0558 ","0559 ","055A ","055B ","055C ","055D ","055E ","055F ",
"0560 ","0561 ","0562 ","0563 ","0564 ","0565 ","0566 ","0567 ",
"0568 ","0569 ","056A ","056B ","056C ","056D ","056E ","056F ",
"0570 ","0571 ","0572 ","0573 ","0574 ","0575 ","0576 ","0577 ",
"0578 ","0579 ","057A ","057B ","057C ","057D ","057E ","057F "},
{"0700 ","Time ","Impul","Suppl","Zone ","0705 ","0706 ","0707 ",  // Miscellaneous
"0708 ","0709 ","070A ","070B ","070C ","070D ","070E ","070F ",
"0710 ","0711 ","0712 ","0713 ","0714 ","0715 ","0716 ","0717 ",
"0718 ","0719 ","071A ","071B ","071C ","071D ","071E ","071F ",
"0720 ","0721 ","0722 ","0723 ","0724 ","0725 ","0726 ","0727 ",
"0728 ","0729 ","072A ","072B ","072C ","072D ","072E ","072F ",
"0730 ","0731 ","0732 ","0733 ","0734 ","0735 ","0736 ","0737 ",
"0738 ","0739 ","073A ","073B ","073C ","073D ","073E ","073F ",
"OOBNB","0741 ","0742 ","0743 ","0744 ","0745 ","0746 ","0747 ",
"0748 ","0749 ","074A ","074B ","074C ","074D ","074E ","074F ",
"0750 ","0751 ","0752 ","0753 ","0754 ","0755 ","0756 ","0757 ",
"0758 ","0759 ","075A ","075B ","075C ","075D ","075E ","075F ",
"0760 ","0761 ","0762 ","0763 ","0764 ","0765 ","0766 ","0767 ",
"0768 ","0769 ","076A ","076B ","076C ","076D ","076E ","076F ",
"0770 ","0771 ","0772 ","0773 ","0774 ","0775 ","0776 ","0777 ",
"0778 ","0779 ","077A ","077B ","077C ","077D ","077E ","077F "},
{"0900 ","0901 ","0902 ","0903 ","0904 ","0905 ","0906 ","0907 ",  // Public Service
"0908 ","0909 ","090A ","090B ","090C ","090D ","090E ","090F ",
"0910 ","0911 ","0912 ","0913 ","0914 ","0915 ","0916 ","0917 ",
"0918 ","0919 ","091A ","091B ","091C ","091D ","091E ","091F ",
"0920 ","0921 ","0922 ","0923 ","0924 ","0925 ","0926 ","0927 ",
"0928 ","0929 ","092A ","092B ","092C ","092D ","092E ","092F ",
"0930 ","0931 ","0932 ","0933 ","0934 ","0935 ","0936 ","0937 ",
"0938 ","0939 ","093A ","093B ","093C ","093D ","093E ","093F ",
"0940 ","0941 ","0942 ","0943 ","0944 ","0945 ","0946 ","0947 ",
"0948 ","0949 ","094A ","094B ","094C ","094D ","094E ","094F ",
"0950 ","0951 ","0952 ","0953 ","0954 ","0955 ","0956 ","0957 ",
"0958 ","0959 ","095A ","095B ","095C ","095D ","095E ","095F ",
"0960 ","0961 ","0962 ","0963 ","0964 ","0965 ","0966 ","0967 ",
"0968 ","0969 ","096A ","096B ","096C ","096D ","096E ","096F ",
"0970 ","0971 ","0972 ","0973 ","0974 ","0975 ","0976 ","0977 ",
"0978 ","0979 ","097A ","097B ","097C ","097D ","097E ","097F "},
{"0B00 ","WSCod","WSMes","0B03 ","0B04 ","0B05 ","0B06 ","0B07 ",  // Reserved
Class
"0B08 ","0B09 ","0B0A ","0B0B ","0B0C ","0B0D ","0B0E ","0B0F ",
"0B10 ","0B11 ","0B12 ","0B13 ","0B14 ","0B15 ","0B16 ","0B17 ",
"0B18 ","0B19 ","0B1A ","0B1B ","0B1C ","0B1D ","0B1E ","0B1F ",
"0B20 ","0B21 ","0B22 ","0B23 ","0B24 ","0B25 ","0B26 ","0B27 ",
"0B28 ","0B29 ","0B2A ","0B2B ","0B2C ","0B2D ","0B2E ","0B2F ",
"0B30 ","0B31 ","0B32 ","0B33 ","0B34 ","0B35 ","0B36 ","0B37 ",
"0B38 ","0B39 ","0B3A ","0B3B ","0B3C ","0B3D ","0B3E ","0B3F ",
"0B40 ","0B41 ","0B42 ","0B43 ","0B44 ","0B45 ","0B46 ","0B47 ",
"0B48 ","0B49 ","0B4A ","0B4B ","0B4C ","0B4D ","0B4E ","0B4F ",
"0B50 ","0B51 ","0B52 ","0B53 ","0B54 ","0B55 ","0B56 ","0B57 ",
"0B58 ","0B59 ","0B5A ","0B5B ","0B5C ","0B5D ","0B5E ","0B5F ",
"0B60 ","0B61 ","0B62 ","0B63 ","0B64 ","0B65 ","0B66 ","0B67 ",
"0B68 ","0B69 ","0B6A ","0B6B ","0B6C ","0B6D ","0B6E ","0B6F ",
"0B70 ","0B71 ","0B72 ","0B73 ","0B74 ","0B75 ","0B76 ","0B77 ",
"0B78 ","0B79 ","0B7A ","0B7B ","0B7C ","0B7D ","0B7E ","0B7F "},
{"0D00 ","0D01 ","0D02 ","0D03 ","0D04 ","0D05 ","0D06 ","0D07 ",  // Undefined Class
"0D08 ","0D09 ","0D0A ","0D0B ","0D0C ","0D0D ","0D0E ","0D0F ",
"0D10 ","0D11 ","0D12 ","0D13 ","0D14 ","0D15 ","0D16 ","0D17 ",
"0D18 ","0D19 ","0D1A ","0D1B ","0D1C ","0D1D ","0D1E ","0D1F ",
"0D20 ","0D21 ","0D22 ","0D23 ","0D24 ","0D25 ","0D26 ","0D27 ",
"0D28 ","0D29 ","0D2A ","0D2B ","0D2C ","0D2D ","0D2E ","0D2F ",
"0D30 ","0D31 ","0D32 ","0D33 ","0D34 ","0D35 ","0D36 ","0D37 ",
"0D38 ","0D39 ","0D3A ","0D3B ","0D3C ","0D3D ","0D3E ","0D3F ",
"0D40 ","0D41 ","0D42 ","0D43 ","0D44 ","0D45 ","0D46 ","0D47 ",
"0D48 ","0D49 ","0D4A ","0D4B ","0D4C ","0D4D ","0D4E ","0D4F ",
"0D50 ","0D51 ","0D52 ","0D53 ","0D54 ","0D55 ","0D56 ","0D57 ",
"0D58 ","0D59 ","0D5A ","0D5B ","0D5C ","0D5D ","0D5E ","0D5F ",
"0D60 ","0D61 ","0D62 ","0D63 ","0D64 ","0D65 ","0D66 ","0D67 ",
"0D68 ","0D69 ","0D6A ","0D6B ","0D6C ","0D6D ","0D6E ","0D6F ",
"0D70 ","0D71 ","0D72 ","0D73 ","0D74 ","0D75 ","0D76 ","0D77 ",
"0D78 ","0D79 ","0D7A ","0D7B ","0D7C ","0D7D ","0D7E ","0D7F "}};

void (*XDSDisp[CLASSMAX][RECORDMAX])() = {
 {Def, Def, Def, Def, Typ, Vcp, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, // Current
Class
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def},
 {Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,  // Future
Class
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def},
 {Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,  // Channel
Class
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def},
 {Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,  //
Miscellaneous
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def},
 {Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,  // Public
Service
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def},
 {Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,  //
Reserved
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,},
 {Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,  //
Undefined Class
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def,
  Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def, Def}};

char *mpaaRatings[] = {"N/A", "G", "PG", "PG-13", "R", "NC-17", "X", "Not Rated"};
char *tvpgRatings[] = {"None", "TV-Y", "TV-Y7 ", "TV-G", "TV-PG ", "TV-14 ", "TV-MA
", "None"};
char flagmask[] = { 0, 0, 0x10, 0, 0xf, 0xf, 0xe, 0};

char *types[] = {
  "Education", "Entertainment", "Movie", "News", "Religious", "Sports", "OTHER",
"Action",
  "Advertisement", "Animated", "Anthology", "Automobile", "Awards", "Baseball",
"Basketball", "Bulletin",
  "Business", "Classical", "College", "Combat", "Comedy", "Commentary", "Concert",
"Consumer",
  "Contemporary", "Crime", "Dance", "Documentary", "Drama", "Elementary", "Erotica",
"Exercise",
  "Fantasy", "Farm", "Fashion", "Fiction", "Food", "Football", "Foreign", "Fund Raiser",
  "Game/Quiz", "Garden", "Golf", "Government", "Health", "High School", "History",
"Hobby",
 "Hockey", "Home", "Horror", "Information", "Instruction", "International", "Interview",
"Language",
 "Legal", "Live", "Local", "Math", "Medical", "Meeting", "Military", "Miniseries",
 "Music", "Mystery", "National", "Nature", "Police", "Politics", "Premiere", "Prerecorded",
 "Product", "Professional", "Public", "Racing", "Reading", "Repair", "Repeat", "Review",
 "Romance", "Science", "Series", "Service", "Shopping", "Soap Opera", "Special",
"Suspense",
 "Talk", "Technical", "Tennis", "Travel", "Variety", "Video", "Weather", "Western"};

char *CType[] = {"Pop    ","Paint  ", "Roll2  ", "Roll3  ", "Roll4  "};
```

ExtractDB

File: hexdec.pl

```
#!/usr/bin/perl

## hex dec display

$a=chr(32);
print ">".$a."<"."\n";

$a=hex(32);
print ">".$a."<"."\n";


$a=hex(20);

print "\n";
#print pack"h2","32";

$a = sprintf "%lx",(32);

print $a;
```

File: stamp2db.pl

```
#!/usr/bin/perl

# this program access mysql DB and reads off all
#or as much data as you want,
#it then generates a web page to view it in raw form.

use DBI;

my($dbh, $sth);

  $dbh = DBI->connect( 'DBI:mysql:caption:localhost:3306', 'root', '') || warn "connecting:
$DBI::errstr\n";


$sth = $dbh->prepare("select * from raw order by recordnum");
$sth->execute();

# get all from DB


while(@row = $sth->fetchrow()) {
#       print @row[2]."\n";
 $tim=@row[1];
 $_=@row[2];
 $l1=0;$l2=0;
 if(/^(\d)\s(\d+)\,\s*$/) {$fnum=$1; $l1=$2;}
 elsif(/^(\d)\s(\d+)\,(\d+)\,\s*$/) {$fnum=$1; $l1=$2; $l2=$3; }
 #else {print "OPPS".@row[2]."\n";}
 else {}


 if( ($l1 == 20 || $l1 == 28)  && $l2 == 44 ) {}

 else
 {

   if(1 == ($fnum+0)) {
   #print $tim;
   if ($l1 >= 0x20  && $l1 < 0x7f) {$out.=chr($l1);}
   elsif($l1 != 0) {$a = sprintf "%lx",$l1+0;
         $out.="[".$a."]";}
   if ($l2 >= 0x20  && $l2 < 0x7f) { $out.=chr($l2);}
```

```
          elsif($12 != 0) { $a = sprintf "%lx",$12+0;
          $out.="[".$12."]";}
      #print "\n";
    }

} # end while


if(2 == ($fnum+0)) {
  if ($11 >= 0x20  && $11 < 0x7f) {$out2.=chr($11);}
          elsif($11 != 0) {$out2.="[".$11."]";}
  if ($12 >= 0x20  && $12 < 0x7f) {$out2.=chr($12);}
          elsif($12 != 0) {$out2.="[".$12."]";}
}



   }


# write out web page with data

open(FILE,"> /usr/www/names/docs/caption_new.html");
print FILE "<HTML>Strings or repeatitive [20], or [28], are not shown<br><br><font
color=blue>Field 1</font><br><br>\n";
$_=$out;


# do necessary substitution to flag URL
s/\<([^\<\>]*)\>/\[$1\]/g;

s/\[20\]/\x04/g;
s/(\[28\]\*)([^\x04]*)(\x04)/\<font color=red\>$1$2$3\<\/font\>/g;
s/\x04/\[20\]/g;
$out=$_;

print FILE $out;

print FILE "<br><br><font color=blue>Field 2</font><br><br>\n";

print FILE $out2;

close(FILE);
```

Release

File: funct.obj

```
L_.ÿl™9D_,.drectve(D_
_.text_l__P`.debug$F_‰_™__H__B.text;_£_Þ___P`.data__       @_0À.data__
        @_0À.debug$F__     /          _H__B.text;_9t
__P`.data!œ
@_0À.debug$F_½
Í
_H__B.text,×
Y___P`.debug$F_c_s__H__B.textQ_}_Î
__P`.data_d_@_0À.data_|_@_0À.data_•_@_0À.debug$F_-_½__H__B.text_Ç_
_P`.debug$F_ß_ï__H__B.textEù__P`.debug$F_>_N__H__B.textIX_
_P`.debug$F_¡_±__H__B.textl»__P`.debug$F_'_7__H__B.textCA_
_P`.debug$F_„_”__H__B.textIž__P`.debug$F_ç_÷__H__B.text2__
_P`.debug$F_3_C__H__B.text_M__P`.debug$F_]_m__H__B.text7w_
_P`.debug$F_®_¾__H__B.textB_È_
___P`.data_ú_@_0À.data_ÿ_@_0À.data___@_0À.data_
        _@_0À.data___@_0À.data___@_0À.data___@_0À.data_
_@_0À.data_%_@_0À.debug$F_*_:__H__B-defaultlib:LIBCMT -defaultlib:OLDNAMES
3À‹L$_ŠD$_H™+ÂÁø_‰□H_,_Ã____‹T$_ƒì_SVW3ÛŠ\$_U‹,H_‰šL_Áà__ÃÁà_□@ƒ
¼_P___…˜‹l$ ‹
,‰MŠ@_ˆE_‹,H_Áà__,L_□@+À□<M¹ÿÿÿò®÷Ñ+ù‰L$_‹÷¹ÿÿÿ‹ý+Àò®O‹L$_Áé_ó¥‹
L$_ƒá_ó¤¿¹ÿÿÿ+Àò®÷Ñ+ù‰L$_‹÷¹ÿÿÿ‹ý+Àò®O‹L$_Áé_ó¥‹L$_ƒá_ó¤‹,H_Áà__,L_Áà
_□@Ç„_P__‹,H_Áà__,L_Áà_□@Ç„_T_‹ŠH_□_KÁá_@_ŠL_]Áá__^□_I‰„_|_,_[ƒÄ_Ã<
__A__e__™__ Error: OverwriteXDS
;______‹T$_ƒì_SVW3ÛŠ\$_U‹,H_‰šL_Áà__ÃÁà_□@ƒ¼_P___…ø‹l$ ‹
,‰MŠ@_ˆE_‹,H_Áà__,L_□@+À□<M¹ÿÿÿò®÷Ñ+ù‰L$_‹÷¹ÿÿÿ‹ý+Àò®O‹L$_Áé_ó¥‹
L$_ƒá_ó¤¿¹ÿÿÿ+Àò®÷Ñ+ù‰L$_‹÷¹ÿÿÿ‹ý+Àò®O‹L$_Áé_ó¥‹L$_ƒá_ó¤‹,H_Áà__,L_Áà
_□@Ç„_P__‹,H_Áà__,L_Áà_□@Ç„_T_‹ŠH_□_KÁá_@_ŠL_Áá_□_I‰„_|_,_]_^[ƒÄ_Ã<
__A__e__™__ Error: Continue with no
Start;______‹T$_SVW‹,H_3ÛÁà_ŠL$__,L_ŠÙÁà_□4@_œ_|_‹²H_Áæ__²L_‹ÆÁà__¾¶□
<@‹„_T_;ð~ _ÂˆŒ8X_‹,H__Áà_^_,L_[Áà_□@,_ÿ„_T_ÃB!_,___
_‹L$_SVW‹□H_Áà__□L_Áà_□_@Ç„
P_‹±H_Áæ__±L_‹ÆÁà_□_@_¾†□<
‹”
T_;Â_Ž“‹T$_¡¾‰__Š^_ˆZ_‹□H_Áà__□L_□@+À□<M¹ÿÿÿò®÷Ñ+ù‹Ù‹÷¹ÿÿÿ‹ú+Àò®O‹
ËÁé_ó¥‹Ëƒá_ó¤¿¹ÿÿÿ+Àò®÷Ñ+ù‹Ù‹÷¹ÿÿÿ‹ú+Àò®O‹ËÁé_ó¥‹Ëƒá_ó¤,__^[Ã_¾†;Â_□”‹
T$_‹5,‰2Š@_ˆB_‹□H_Áà__□L_□@+À□<M¹ÿÿÿò®÷Ñ+ù‹Ù‹÷¹ÿÿÿ‹ú+Àò®O‹ËÁé_ó¥‹
Ëƒá_ó¤¿¹ÿÿÿ+Àò®÷Ñ+ù‹Ù‹÷¹ÿÿÿ‹ú+Àò®O‹ËÁé_ó¥‹Ëƒá_ó¤,__^[Ã3ÀŠD$__‡|_ƒÄ_¯▒_
„”‹T$_‹5,‰2Š@_ˆB_‹□H_Áà__□L_□@+À□<M¹ÿÿÿò®÷Ñ+ù‹Ù‹÷¹ÿÿÿ‹ú+Àò®O‹ËÁé_
ó¥‹Ëƒá_ó¤¿¹ÿÿÿ+Àò®÷Ñ+ù‹Ù‹÷¹ÿÿÿ‹ú+Àò®O‹ËÁé_ó¥‹Ëƒá_ó¤,__^[Ã‹T$_QRÿ_µƒÄ_,
__^[ÃA1_\__a__„__²/_í!_ÿ______'___U_,_ª___¯___Ò___)_A_0_ Error: bad check sum
Error: too many bytes Error: too few
bytesQ____&_3Ò‹D$_ŠT$_‹L$_‰T□ü,_Ã__6_‹L$_SVI‹t$_□_M_TŽ_□_Ž_Ò_P_□Œ-
__‹_=ÿ}_Áâ_Š\$__Ö^\__ÿ_,_^[ÃE___;_‹T$_‹L$_□D'üJÇ@_ƒ8_}_Ç@_Ç„Ñ8_,_ÃÇ@__Ç
„Ñ<_,_ÃI_@_‹L$_VWI‹t$_¿□_ŽÇ@_ƒ8_|_¿_ŠT$_‰x_€ú%u_□_OÇ„†8__€ú&u_□_OÇ„
†8__€ú'u_□_OÇ„†8__,__^Ãl___E_‹T$_VJ‹L$_3À□4'‰D'_ƒ>_}_‰F_,_Ç„Ñ8__^Ã,_‰F
_^‰„Ñ<_,_ÃC___J_‹T$_‹D$_□L□üJÇA_ƒ9_}_ÇA_Ç„Ð8_,_ÃÇA__Ç„Ð<_,_ÃI_O_‹L$_
‹D$_□D^üÇ@__ƒ8_}
Ç@_,_ÃÇ@__,_Ã2_T_‹L$_,_‰A_‰A_Ã__Y_‹L$_ŠD$_I< r$<?w ‹D$_□_^ƒ8_}
Ç@_,_ÃÇ@__,_Ã7_^_‹T$_ƒì_S□Jÿ‹\$_VWU‹t‹_‹|‹_□_‹□_O□,Fƒ¼«___Ž__ƒú__…U_…
ö_…°…ÿuV‹T$ ‹
¾‰
ŠN_^J_¹ÿÿÿÿ‹„ƒ8_‹<…+Àò®÷Ñ+ù‰L$_‹÷¹ÿÿÿÿ‹ú+Àò®O‹L$_Áé_ó¥‹L$_éA_‹T$
‹5¹‰2ŠI_^J_¹ÿÿÿÿ‹„ƒ8_‹<…+Àò®÷Ñ+ù‰L$_‹÷¹ÿÿÿÿ‹ú+Àò®O‹L$_Áé_ó¥‹L$_éë_…ÿuL‹
T$ ¿¡¹‰_+ÀŠI_^J_¹ÿÿÿÿò®÷Ñ+ù‰L$_‹÷¹ÿÿÿÿ‹ú+Àò®O‹L$_Áé_ó¥‹L$_é›_‹T$ ¿‹
,‰
¹ÿÿÿÿŠ@_^B_+Àò®÷Ñ+ù‰L$_‹÷¹ÿÿÿÿ‹ú+Àò®O‹L$_Áé_ó¥‹L$_éN_…ö_…°…ÿuV‹T$
‹5¹‰2ŠI_^J_¹ÿÿÿÿ‹„ƒ8_‹<…+Àò®÷Ñ+ù‰L$_‹÷¹ÿÿÿÿ‹ú+Àò®O‹L$_Áé_ó¥‹L$_éì‹T$
‹5¹‰2ŠI_^J_¹ÿÿÿÿ‹„ƒ8_‹<…+Àò®÷Ñ+ù‰L$_‹÷¹ÿÿÿÿ‹ú+Àò®O‹L$_Áé_ó¥‹L$_é–
…ÿuJ‹T$ ¿‹
,‰
¹ÿÿÿÿŠ@_^B_+Àò®÷Ñ+ù‰L$_‹÷¹ÿÿÿÿ‹ú+Àò®O‹L$_Áé_ó¥‹L$_ëH‹T$ ¿‹
,‰
¹ÿÿÿÿŠ@_^B_+Àò®÷Ñ+ù‰L$_‹÷¹ÿÿÿÿ‹ú+Àò®O‹L$_Áé_ó¥‹L$_ƒá_ó¤‹Å‹Œ«__Áà__ËÆ
D__□|__¹ÿÿÿ+Àò®÷Ñ+ù‰L$_‹÷¹ÿÿÿÿ‹ú+Àò®O‹L$_Áé_ó¥‹L$_ƒá_ó¤Ç„«__,_]_^[ƒÄ_Ã
L~_Q~_l▒_¢{_§{_Â▒_ûu__x___x_G_u_M_r_R_r_¡_o_¦_o_Á_▒_÷_l_ü_l___▒_P_u_V_i_
[_i_š_u__f_¥_f_T4  T3  C4  C3  T2        T1  C2  C1
B_____c_.fileþÿg_J:\Vb60\Line21\funct.c.drectve___(.text_______
_.debug$F_______.text___;_____
_.data________.data_____E___XDSMes_.debug$F_______.text___;____XDSCont__.data
         __!_]  _.debug$F
______.text___,___XDSData__–_.debug$F______.text
__Q____XDSEnd
 _.data_____¡__.data_____Ô__.data__________XDSDisp_;__.debug$F_____
_.text_____F___.debug$F_______.text___E_S___.debug$F_______.text___I_]__
_.debug$F_______.text___l_j___.debug$F_______.text___C_x__
_.debug$F_______.text___I_‡___.debug$F_______.text___2_›___.debug$F_______.text
____§__.debug$F!_____.text"__7_²_"_.debug$F#____"_.text$__B___Á_$
_.data%____Ë_%_.data&____ä_&_.data'____ý_'_.data(______(_.data)____/_)_.data*____H
_*_.data+____h_+_.data,____□_,_.data-____š_-
__CType_.debug$F.____$_³__XDSBegin_XDSStart??_C@_0BE@FLOP@?5?5Error?3?5?
5Overwrite?$AA@??_C@_04EPNH@XDS?5?$AA@??_C@_0CB@OHLK@?5?5Error?3?
5?5Continue?5with?5no?5Start@_MaxLength??_C@_0BI@LPLM@?5?5Error?3?5?5bad?5
check?5sum?$AA@??_C@_0BJ@ICAC@?5?5Error?3?5?5too?5many?5bytes?$AA@??_C
@_0BI@NPMA@?5?5Error?3?5?5too?5few?5bytes?$AA@_MinLength_ControlSave_Data
Save_PopOnDecode_RollUpDecode_PaintOnDecode_EndOfCaptionDecode_TextDecode_U
RLDecode_ChannelChange_DumpText??_C@_04DILL@T4?5?5?$AA@??_C@_04IGNK
@T3?5?5?$AA@??_C@_04JHCE@C4?5?5?$AA@??_C@_04CJEF@C3?5?5?$AA@??_C
```

@_04NEMM@T2?5?5?$AA@??_C@_07NDP@?5?5?5?5?5?5?5?$AA@??_C@_04CCPH
@T1?5?5?$AA@??_C@_04HLFD@C2?5?5?$AA@??_C@_04INGI@C1?5?5?$AA@

File: L21Dec.obj

```
L___m™9”_!.drectvelÌ_
_.text_8_ _P`.debug$F_@_P__H__B.text¨Z____ _P`.debug$F_p_€__H__B.textzŠ___
_P`.debug$F_,_<__H__B.text
F_ _P`.debug$F_S_c__H__B.text
m_ _P`.debug$F_z_Š__H__B-defaultlib:LIBCMT -defaultlib:OLDNAMES -
export:_Line21Decode@12 -export:_ShiftRight@8 -export:_ShiftLeft@8 ,_Â__
        _‹T$_SV3ÉW…ÒupÇ_,£P°‹_‰,T°‹_‰J_‹_‰J‹_3ö,_‰J_‹_‰J_°€3ÿ‹_ƒÆ0J‰¼3
_uíHuã3À_^[Âƒú_t_ƒú_u_ŠL$_‹D$_€á□PQÆRèƒÄ3À_^[Â__________#_/__8__A__Q
__a__™__¨_____S¡V3Û‹t$WŠ\$_U‹¬°L°‹|$_‹Å‹
ƒÅ_QWSVÿ_ƒÄ_…Àu_‹ÅƒÅ_ƒ8uöƒÅ_ëÙƒ}t_‹Å‹
ƒÅ_QWSVÿ_ƒÄ_ƒ}uæ‹E_‹
]_‰„±L°^[Ã___"__O__j__z_____‹D$_ŠL$_ÓèÂ_
___‹D$_ŠL$_ÓàÂ_
___.filebÿg_J:\Vb60\Line21\L21Dec.c.drectve___1_CCstateX°____.text______
_.debug$F_______.text___¨_____,_.debug$F_______.text___z__7_
_.debug$F_______.text___
_E_ _.debug$F          ______.text
__
_S
_.debug$F_____
_`_cStatePtr_DllMain@12_Line21Decode@12_DataState_stateMachine_ShiftRight@8_Shif
tLeft@8
```

File: Line21.dll

DYNAMIC LINK LIBRARY
32bit for Windows 95 and Windows NT

Technical File Information:

Image File Header

Signature:	00004550
Machine:	Intel 386
Number of Sections:	0005
Time Date Stamp:	39996d0e
Symbols Pointer:	00000000
Number of Symbols:	00000000
Size of Optional Header	00e0
Characteristics:	File is executable (i.e. no unresolved external references). Line numbers stripped from file. Local symbols stripped from file. 32 bit word machine. File is a DLL.

Image Optional Header

Magic:	010b
Linker Version:	3.00
Size of Code:	00003800
Size of Initialized Data:	00012000
Size of Uninitialized Data:	00000000
Address of Entry Point:	00002680
Base of Code:	00001000
Base of Data:	00005000
Image Base:	10000000
Section Alignment:	00001000
File Alignment:	00000200
Operating System Version:	4.00
Image Version:	0.00
Subsystem Version:	4.00
Reserved1:	00000000
Size of Image:	00019000
Size of Headers:	00000400
Checksum:	00000000
Subsystem:	Image runs in the Windows GUI subsystem.

```
             DLL Characteristics:  0000
            Size of Stack Reserve:  00100000
             Size of Stack Commit:  00001000
             Size of Heap Reserve:  00100000
              Size of Heap Commit:  00001000
                     Loader Flags:  00000000
           Size of Data Directory:  00000010
Export Directory Virtual Address:  00005060
            Export Directory Size:  00000073
Import Directory Virtual Address:  00017000
            Import Directory Size:  00000028
            Base Relocation Table
                  Virtual Address:  00018000
       Base Relocation Table Size:  00000c40
```

Export Table

```
______________________Name:  Line21.dll
           Characteristics:  00000000
           Time Date Stamp:  39996d0e
                   Version:  0.00
                      Base:  00000001
       Number of Functions:  00000003
           Number of Names:  00000003

           Ordinal  Entry Point  Name
           0000     00001010     Line21Decode
           0001     00001150     ShiftLeft
           0002     00001140     ShiftRight
```

Import Table

```
KERNEL32.dll
Ordinal      Function Name

0092  GetACP
0103  GetProcAddress
00eb  GetModuleHandleA
0137  GetVersion
0062  ExitProcess
00c7  GetCurrentThreadId
0222  TlsSetValue
021f  TlsAlloc
```

```
0220   TlsFree
01fd   SetLastError
0221   TlsGetValue
00e1   GetLastError
0155   HeapCreate
0157   HeapDestroy
01fa   SetHandleCount
00dc   GetFileType
0116   GetStdHandle
0114   GetStartupInfoA
0044   DeleteCriticalSection
00e9   GetModuleFileNameA
0098   GetCPInfo
009f   GetCommandLineA
00f6   GetOEMCP
008b   FreeEnvironmentStringsA
00d0   GetEnvironmentStrings
008c   FreeEnvironmentStringsW
00d2   GetEnvironmentStringsW
0242   WideCharToMultiByte
024f   WriteFile
0164   InitializeCriticalSection
004f   EnterCriticalSection
0177   LeaveCriticalSection
0153   HeapAlloc
0159   HeapFree
0178   LoadLibraryA
0083   FlushFileBuffers
0016   CloseHandle
0206   SetStdHandle
01f8   SetFilePointer
```

Section Table

```
_____________Section name:   .text
              Virtual Size:   0000379b
           Virtual Address:   00001000
         Size of raw data:   00003800
      Pointer to Raw Data:   00000400
   Pointer to Relocations:   00000000
  Pointer to Line Numbers:   00000000
   Number of Relocations:   0000
  Number of Line Numbers:   0000
          Characteristics:   Section contains code
                             Section is executable
```

```
                          Section is readable


            Section name:  .rdata
            Virtual Size:  000000d3
         Virtual Address:  00005000
        Size of raw data:  00000200
     Pointer to Raw Data:  00003c00
  Pointer to Relocations:  00000000
 Pointer to Line Numbers:  00000000
  Number of Relocations:   0000
 Number of Line Numbers:   0000
         Characteristics:  Section contains initialized data
                           Section is readable


            Section name:  .data
            Virtual Size:  00010648
         Virtual Address:  00006000
        Size of raw data:  00004400
     Pointer to Raw Data:  00003e00
  Pointer to Relocations:  00000000
 Pointer to Line Numbers:  00000000
  Number of Relocations:   0000
 Number of Line Numbers:   0000
         Characteristics:  Section contains initialized data
                           Section is readable
                           Section is writeable


            Section name:  .idata
            Virtual Size:  00000420
         Virtual Address:  00017000
        Size of raw data:  00000600
     Pointer to Raw Data:  00008200
  Pointer to Relocations:  00000000
 Pointer to Line Numbers:  00000000
  Number of Relocations:   0000
 Number of Line Numbers:   0000
         Characteristics:  Section contains initialized data
                           Section is readable
                           Section is writeable
```

09651405.050904

Section name: .reloc
Virtual Size: 00000ed6
Virtual Address: 00018000
Size of raw data: 00001000
Pointer to Raw Data: 00008800
Pointer to Relocations: 00000000
Pointer to Line Numbers: 00000000
Number of Relocations: 0000
Number of Line Numbers: 0000
Characteristics: Section contains initialized data
Section can be discarded
Section is readable

Header Information

________________*Signature:* 5a4d
Last Page Size: 0090
Total Pages in File: 0003
Relocation Items: 0000
Paragraphs in Header: 0004
Minimum Extra Paragraphs: 0000
Maximum Extra Paragraphs: ffff
Initial Stack Segment: 0000
Initial Stack Pointer: 00b8
Complemented Checksum: 0000
Initial Instruction Pointer: 0000
Initial Code Segment: 0000
Relocation Table Offset: 0040
Overlay Number: 0000
Reserved: 0000 0000 0000 0000
0000 0000 0000 0000
0000 0000 0000 0000
0000 0000 0000 0000
Offset to New Header: 00000080
Memory Needed: 2K

File: Line21.exp

```
L___m™9R___.edatasdØ
@@.drectve_<_
_m™9F___(4@Q^h__Line21.dllLine21DecodeShiftLeftShiftRight___ _$_(__4_,__8_0__<_
/DLL
/OUT:Line21.dll.edata__&_E_7_ÜLine21DecodeShiftRightShiftLeft_Line21Decode@12_S
hiftRight@8_ShiftLeft@8__imp__Line21Decode@12__imp__ShiftLeft@8__imp__ShiftRig
ht@8__IMPORT_DESCRIPTOR_Line21__NULL_IMPORT_DESCRIPTOR▒Line21_NU
LL_THUNK_DATA
```

File: Line21.lib

```
!<arch>
/           966356238           0      222       `
         _p_˜
,_R_Þ_p_˜
,_š_Line21Decode@12_ShiftLeft@8_ShiftRight@8__IMPORT_DESCRIPTOR_Line21__N
ULL_IMPORT_DESCRIPTOR__imp__Line21Decode@12__imp__ShiftLeft@8__imp__Sh
iftRight@8▒Line21_NULL_THUNK_DATA/           966356238           0      244       `
         R_Þ_š_p_,
˜
¯           __________Line21Decode@12_ShiftLeft@8_ShiftRight@8__IMPORT_DESCRIPT
OR_Line21__NULL_IMPORT_DESCRIPTOR__imp__Line21Decode@12__imp__ShiftLef
t@8__imp__ShiftRight@8▒Line21_NULL_THUNK_DATALine21.dll/    966356238
0     592       `
L___m™9,__à____________.idata$2_D_X__@_À.idata$6v_X_@
À_______Line21.dll___.idata$2@À_h.idata$6__.idata$4@Àh.idata$5@Àh__8_P__IMPOR
T_DESCRIPTOR_Line21__NULL_IMPORT_DESCRIPTOR▒Line21_NULL_THUNK_D
ATALine21.dll/    966356238           0      127       `
L___m™9P__.idata$3_<@_À______NULL_IMPORT_DESCRIPTOR
Line21.dll/    966356238           0      154       `
L___m™9l__.idata$5_d@0À.idata$4_h@0À____▒Line21_NULL_THUNK_DATALine21.
dll/    966356238           0      492       `
L___m™9ð__.text_'°_ _ `.idata$5_ÄÈ_@_0À.idata$4_ÒÖ_@_0À.idata$6_à@_
Àÿ%_______Line21Decode.text______□`Â

¯.idata$5______□`Â___.idata$4_______□`Â.idata$6______□`Â__6__imp__Line21Decode
@12__IMPORT_DESCRIPTOR_Line21Line21.dll/    966356238           0      484       `
L___m™9ì__.text_'°_ _ `.idata$5_ÄÈ_@_0À.idata$4_ÒÖ_@_0À.idata$6à@_
Àÿ%________ShiftLeft.text______□`Â

¯.idata$5______□`Â___.idata$4_______□`Â.idata$6______□`Â__2__imp__ShiftLeft@8__I
MPORT_DESCRIPTOR_Line21Line21.dll/    966356238           0      487       `
L___m™9î__.text_'°_ _ `.idata$5_ÄÈ_@_0À.idata$4_ÒÖ_@_0À.idata$6_à@_
Àÿ%________ShiftRight.text______□`Â

¯.idata$5______□`Â___.idata$4_______□`Â.idata$6______□`Â__3__imp__ShiftRight@8
__IMPORT_DESCRIPTOR_Line21
```

File: Line21.pch

```
VCPCH0…z&□Xü
Sep 27 199513:24:33š_d________¯_ÿÿÿÿ_____ ___j:\vb60\line21\vc40.pdb___†_Q
0_I___INTEGRAL_MAX_BITS=64_
_M_IX86=400___MSC_VER=1000___WIN32___MT__WIN32__NDEBUG_
_WINDOWS__J:\VB60\Line21__C:\MSDEV\INCLUDE__C:\MSDEV\MFC\include__K:\
MSDEV\INCLUDE__K:\MSDEV\MFC\include__D:\MSDEV\INCLUDE__D:\MSDEV\MF
C\include
        ________¯_ÿÿÿÿ_____ ___j:\vb60\line21\vc40.pdb__p™€œ™€È™€Är€Ä_€Äb€_€€€
£€Ø¢€èC€Ä2€D€DB€øA€ÈA€tB€€B€ÄB€ìA€ØA€”B€_B€,>€´B€$B€4B€`B€_|€0|€_|€Ì{€ð
z€_{€Üz€,{€@{€T{€h{€|{€□{€¤{€,{€à{€,{€@{€Üz€_{€_{€Èz€ô{€˜Ž€¼Ž€@|€xŽ€$€D€d
€„€¤€Ä€ä€__€$_€D_€d_€„_€¤_€Ä_€ä_€__€$_€D_€d_€„_€¤_€Ä_€ä_€__€$_€D_€d_€„_€¤
_€Ä_€ä_€__€$_€D_€d_€„_€¤_€Ä_€ä_€__€$_€D_€d_€„_€¤_€Ä_€ä_€__€$_€D_€d_€„_€¤_
€Ä_€ä_€__€$_€D_€d_€„_€¤_€Ä_€ä_€__€$_€D_€d_€„_€¤_€Ä_€ä_€_    €$      €D      €d
      €„      €¤      €Ä      €ä      €_
€$
€D
€d
€„
€¤
€Ä
€ä
€__€$_€D_€d_€„_€¤_€Ä_€ä_€_€$€D€d€„€¤€Ä€ä€_
€$
€D
€d
€„
€¤
€Ä
€ä
€__€$_€D_€d_€„_€¤_€Ä_€ä_€__€$_€D_€d_€„_€¤_€Ä_€ä_€__€$_€D_€d_€„_€¤_€Ä_€ä_€
__€$_€D_€d_€„_€¤_€Ä_€ä_€__€$_€D_€d_€„_€¤_€ÜB€ÄR€€€__¤€ø[_Iÿÿ?_ÿÿ>_X_____
___
_____è_¼_ Release/Parity.obj_
Àÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ

09951605.050604

ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ
ÿÿ

09851505.050901

```
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿJk0___ÿÿ
_p_€ñ
___€_ñ
_p__ñ
_____ñ
_____ñ___òñ
______

__
_____ñ
_____ñ
_p__ñ
_p__ñ
_p__€_ñ
_p___ñ"_€__MIDL_STUB_MESSAGEñ__
________

________
__f________pfnTranslateToXmitñ_______pfnTranslateFromXmitóòñ_______pfnFreeXmit_
_____pfnFreeInst&_______XMIT_ROUTINE_QUINTUPLE______
_________

__
_____ _òñ
________
__F________pObjóòñ__u___Sizeóòñ_______pfnBind_____
        pfnUnbindòñ&________GENERIC_BINDING_INFOñ
_____ñŽ_______
        reserved1òñ__"___dataRepresentationñ_______Bufferñ__"__cbBufferóòñ__"___iMe
thod______   reserved2òñ__"_(_rpcFlagsóòñ"____,_tagRPCOLEMESSAGEóòñ____u
_____ ___
!_
________
#_&____"___pfnAllocate__$____pfnFree"__%____MALLOC_FREE_STRUCT&________
pfnBind______
        pfnUnbindòñ.__'____GENERIC_BINDING_ROUTINE_PAIRòñ*______
CommOffsetñ______FaultOffset"__)__l0_†k™9___
        _Debug/XDSROM.objF__<Microsoft (R) 32-bit C/C++ Optimizing Compiler
Version 10.00________        MaxLength___˜____  MinLength________XDSMes____,_
        __XDSDisp____:_
__mpaaRatings___8:_
__tvpgRatings___X:____flagmask___`:___types___à;_
```

```
__CType______XMIT_ROUTINE_QUINTUPLE_______GENERIC_BINDING_INFO___
__tagRPCOLEMESSAGE__&___MALLOC_FREE_STRUCT"_(___GENERIC_BINDING
_ROUTINE_PAIR__&__MALLOC_FREE_STRUCT__*___COMM_FAULT_OFFSETS"_
.___FULL_PTR_TO_REFID_ELEMENT"_(__GENERIC_BINDING_ROUTINE_PAIR___
__XMIT_ROUTINE_QUINTUPLE"_,__PFULL_PTR_TO_REFID_ELEMENT
_1__GUID__3__LPCLSID__6__tagENHMETARECORD
_1__GUID__L___MIDL_STUB_DESC__S___FULL_PTR_XLAT_TABLES______PROP
SHEETPAGEA__f___PROPSHEETPAGEW
_1__CLSID__,__IRpcChannelBufferVtbl__„__DLGTEMPLATE__‡___RPC_SYNTAX_ID
ENTIFIER__Œ___EXCEPTION_RECORD__Ž___OVERLAPPED__‘___FLOATING_SA
VE_AREA
_1__IID__„__DLGTEMPLATE_____PROPS_(,ðŽ$_öŽ_,öŽ___HEETPAGEA__’__LPDL
GTEMPLATEA__f__PROPSHEETPAGEW__”__tagPOINT__—
___CONTEXT__™__RPC_VERSION
_”__POINT__—__CONTEXT
_3__LPIID__‰__PEXCEPTION_RECORD__›__val_context
____HDWP
_u__PUINT_____RPC_SS_THREAD_HANDLE__X__CALINFO_ENUMPROCW_____R
PC_AUTH_IDENTITY_HANDLE__6_
ENHMETARECORD&_
_SECURITY_CONTEXT_TRACKING_MODE__X__DATEFMT_ENUMPROCW__t_
PSID_NAME_USE___MIDICALLBACK@20"_!_SECURITY_DESCRIPTOR_CONTROL

____
        EXPR_EVAL__ž_LPFRHOOKPROC__X__CALINFO_ENUMPROCA__¡__PFNC
ALLBACK__X__DATEFMT_ENUMPROCA__¤_ICMENUMPROCW__©_
PQUERYHANDLER
_@_FLOAT__"__LPDWORD_____RPC_AUTHZ_HANDLE__u MMVERSION
____HRSRC___RPC_FORWARD_FUNCTION@20"_®__PTOP_LEVEL_EXCEPTION_F
ILTER
____HCONV

_
_BYTE____LONGLONG__a__LPPROPSHEETPAGEW__±_ICMENUMPROCA__!__LP
WSTR__"_ACCESS_MASK__"_COLORREF__Z__LPPROPSHEETPAGEA__´__PIMAG
E_TLS_CALLBACK_____HWAVEOUT__$_PRPC_RUNDOWN____HENHMETAFILE
_t__PBOOL__‡__RPC_SYNTAX_IDENTIFIER__X__DESKTOPENUMPROCW__"_u_lo
ng
_t__SPTR__!__LPOLESTR____NDR_CCONTEXT
_u_UINT__X__WNDENUMPROC__X__DESKTOPENUMPROCA__¶__WNDPROC__"_
HREFTYPE__#        DWORDLONG
_p__LPSTR__t
TOKEN_TYPE
_!__BSTR____
RPC_NS_HANDLE
____HDESK__"__PDWORD__°__EXCEPTION_POINTERS__!
CLIPFORMAT__!__LPWORD__t_TYPEKIND__X__CODEPAGE_ENUMPROCW
_p__HPSTR__d__LPFNPSPCALLBACKW__!_OLECHAR__X__ENUMRESLANGPROC
__t_VARKIND__X__CODEPAGE_ENUMPROCA__]__LPFNPSPCALLBACKA__ _
RPC_BUFPTR__X__WINSTAENUMPROCW__½__RPC_AUTH_KEY_RETRIEVAL_FN
__!_   LPCOLESTR____     SC_HANDLE
____HTASK__À__REGISTERWORDENUMPROCW__X__TIMEFMT_ENUMPROCW__
X__WINSTAENUMPROCA
_!_ATOM__!__LPCWSTR__X__DLGPROC__Ã__REGISTERWORDENUMPROCA____
_HOLEMENU
_t_BOOL___RPC_OBJECT_INQ_FN@12
_"_ULONG__X__TIMEFMT_ENUMPROCA__«__PEXCEPTION_POINTERS__Æ__tag
HELPINFO__t_CALLCONV_____LPMENUTEMPLATEA_____HGLOBAL_____LPVOI
D____
RPC_IF_HANDLE__X__HOOKPROC
_p__PSTR__ž_LPCFHOOKPROC__!_USHORT__7_
LPHMIDIOUT____HRESULT__
__PUCHAR____DISPID__È__PFULL_PTR_XLAT_TABLES__ž_LPCCHOOKPROC__
__LPBYTE
_X__PROC
_A_DATE__Đ_
ENHMFENUMPROC
____PSID
_t__LPINT
_Ñ__SNB
_p_CHAR__p__LPCSTR__Ó__SERVER_ROUTINE
___SCODE&_Ø__LPOVERLAPPED_COMPLETION_ROUTINE__Û_
        YIELDPROC__!_VARTYPE_____HWAVEIN__ß__IRpcChannelBuffer__ß__IRpc
ChannelBuffer__ž__LPPAGEPAINTHOOK__t
        OLERENDER__u__LPUINT__X_GOBJENUMPROC____
HMETAFILEPICT__Ñ__LPBSTR
_
_byte__@__PFLOAT__"__PSECURITY_INFORMATION__$__NDR_RUNDOWN_____
LPCVOID__"_SERVICE_STATUS_HANDLE__ž__LPPAGESETUPHOOK
_u_u_int__!_wint_t__â_
LPMMIOPROC____
STUB_THUNK
_!_WCHAR_____HGDIOBJ__X__ENUMRESNAMEPROC____HCONVLIST
____HSZ
_t_INT_____HDDEDATA__å__tagHANDLETABLE__X__FARPROC__X_
PROPENUMPROCW__"_MCIERROR__t__LPBOOL
____PLONG__u_WPARAM
____HMMIO__ž_
LPOFNHOOKPROC__ç__HPROPSHEETPAGE__QUERYHANDLER__X_
PROPENUMPROCA
____HKL_____HACCEL__"
RPCOLEDATAREP___VARIANT_BOOL_____HCURSOR__7__PHANDLE__é_
        _onexit_t__X_
```

```
FONTENUMPROCW__"_LCTYPE__î__RPC_DISPATCH_FUNCTION
_"_CALID___
RPC_STATUS_____HMIDIIN
____HDC__ó__MSGBOXCALLBACK__X_
FONTENUMPROCA_____GENERIC_UNBIND_ROUTINE_____HBITMAP____LPARA
M___
MENUTEMPLATEW__7__LPHMIDISTRM__t_TOKEN_INFORMATION_CLASS___D
RVCALLBACK@20__÷__RPC_MESSAGE___
MENUTEMPLATEA
____HMIDI__ú__PFNPROPSHEETCALLBACK_____PSECURITY_DESCRIPTOR
____HMENU_____HWINSTA___
LCSGAMUTMATCH___RPC_CLIENT_FREE@4__ý__PMIDL_STUB_DESC
___LONG____NDR_SCONTEXT____
ARRAY_INFO__p__va_list__7__LPHMIXER__X_       TIMERPROC__X_
DRAWSTATEPROC____      HINSTANCE__"_FOURCC__
___MIDL_STUB_MESSAGE__"_SECURITY_INFORMATION__X__ENUMRESTYPEP
ROC__å__HANDLETABLE__7_   LPHWAVEIN__
__PFORMAT_STRING__"_CALTYPE__7__LPHANDLE_____RPC_MGMT_AUTHORI
ZATION_FN__!_LANGID"_t_SECURITY_IMPERSONATION_LEVEL__u_SOCKET__
_BOOLEAN__7__RPC_EP_INQ_HANDLE
____HWND___MMIOPROC@16_____LPHANDLER_FUNCTION__t_EXCEPTION_DIS
POSITION__tXLAT_SIDE
____HHOOK_____LPTHREAD_START_ROUTINE__7_
LPHWAVEOUT_____HPALETTE__X__PROPENUMPROCEXW_____HMODULE___W
AVECALLBACK@20___RPC_CLIENT_ALLOC@4_____HANDLE_____HMIXER
_!__PWORD__X__PROPENUMPROCEXA_____HCOLORSPACE__X__EDITWORDBR
EAKPROCA_____GENERIC_BINDING_ROUTINE__!_u_short__7_
LPHMIDIIN__X__LOCALE_ENUMPROCW
_"_DWORD__X__GRAYSTRINGPROC__t_DESCKIND
_7__PHKEY"_®__LPTOP_LEVEL_EXCEPTION_FILTER
_"_LCID_____LPFNADDPROPSHEETPAGE__ _u_char__X__LOCALE_ENUMPROCA
____HRGN__t_ACL_INFORMATION_CLASS__t_SYSKIND___TIMECALLBACK@20_

___
PACCESS_TOKEN_____SC_LOCK__X__LINEDDAPROC_____PHANDLER_ROUTIN
E"_!__PSECURITY_DESCRIPTOR_CONTROL__t_FUNCKIND___
FXPT2DOT30_____LPLONG
____HGLRC_____HMIDIOUT__t
INVOKEKIND____  HMIDISTRM
____HFONT
____HPEN
____HICON
____PVOID____
tagMETARECORD__"_REGSAM__t      OLEUPDATE__u_MCIDEVICEID__
__LPSERVICE_MAIN_FUNCTIONW
_!_WORD____       HMETAFILE__X_  ABORTPROC
```

```
___HDRVR__$__LPSERVICE_MAIN_FUNCTIONA__ž__LPPRINTHOOKPROC__u_si
ze_t___         LCSCSTYPE
_ __PBYTE____LRESULT__)_
MFENUMPROC
_é__FPTR
___HKEY
_ _UCHAR_____PTHREAD_START_ROUTINE__!_wctype_t____I_RPC_HANDLE
___HDROP____MEMBERID
__SHORT__!_wchar_t__u_MMRESULT_____XMIT_HELPER_ROUTINE_____RPC_BI
NDING_HANDLE
_" HIMC_____I_RPC_MUTEX____
      HMIXEROBJ__ž__LPSETUPHOOKPROC__,__RPC_BLOCKING_FUNCTION__
___HBRUSH__X_
SENDASYNCPROC__!FILEOP_FLAGS__/__LPTIMECALLBACK_____handle_t_____H
LOCAL
_t_HFILE__    _Debug/funct.objF__<Microsoft (R) 32-bit C/C++ Optimizing Compiler
Version 10.00.__Ø*_% __3___XDSBegin_____field__ _ch___p__mes___1_
      cStatePtr__.__P >__9_Ê__3___XDSStart_____field__ _ch___p__mes___1_
      cStatePtr__.__È >__9____3___XDSCont_____field__ _ch___p__mes___1_
      cStatePtr__.__@______F__3___XDSData_____field__ _ch___p__mes___1_
      cStatePtr__*__´_E__@_Z__3___XDSEnd_____field__ _ch___p__mes___1_
      cStatePtr__2__0_$__Ÿ__3___ControlSave_____field__ _ch___p__mes___1_
      cStatePtr__.__Ì_½     ,Ã__3___DataSave_____field__ _ch___p__mes___1_
      cStatePtr
_ôÿÿÿt_m
_øÿÿÿt_f
_üÿÿÿt_c__2__T_€    {€__3___PopOnDecode_____field__ _ch___p__mes___1_cStatePtr
_üÿÿÿt_f__2__è_Õ    Ð__3__RollUpDecode_____field__ _ch___p__mes___1_  cStatePtr
_øÿÿÿt_f
_üÿÿÿt_c__2__p_€    {Õ__3__
PaintOnDecode_____field__ _ch___p__mes___1_ cStatePtr
_üÿÿÿt_f__6__ü_€    {U__3___EndOfCaptionDecode_____field__ _ch___p__mes___1_
      cStatePtr
_üÿÿÿt_f__.__€_^    YÕ__3__
TextDecode_____field__ _ch___p__mes___1_      cStatePtr
_üÿÿÿt_f__.__ø_)_$3__3__   URLDecode_____field__ _ch___p__mes___1_
      cStatePtr__2__€_l    g\__3__
ChannelChange_____field__ _ch___p__mes___1_ cStatePtr
_üÿÿÿt_f__.____Ä_   ¿_È__3___DumpText_____field__ _ch___p__mes___1_   cStatePtr
_ôÿÿÿt_m
_øÿÿÿt_f
_üÿÿÿt_c_________ MaxLength___˜____ MinLength_________XDSMes____,_
      __XDSDisp___à;_4__CType______XMIT_ROUTINE_QUINTUPLE_______GENE
RIC_BINDING_INFO_____tagRPCOLEMESSAGE__&___MALLOC_FREE_STRUCT"_(
___GENERIC_BINDING_ROUTINE_PAIR__&__MALLOC_FREE_STRUCT__*___CO
MM_FAULT_OFFSETS"_.___FULL_PTR_TO_REFID_ELEMENT"_(__GENERIC_BIND
ING_ROUTINE_PAIR_____XMIT_ROUTINE_QUINTUPLE"_,__PFULL_PTR_TO_REFI
D_ELEMENT
_1___GUID__3__LPCLSID__6__tagENHMETARECORD
_1__GUID__L___MIDL_STUB_DESC__S___FULL_PTR_XLAT_TABLES______PROP
SHEETPAGEA__f___PROPSHEETPAGEW
_1__CLSID__7__XDSState__,__IRpcChannelBufferVtbl__„__DLGTEMPLATE__‡___RP
C_SYNTAX_IDENTIFIER__Œ___EXCEPTION_RECORD__Ž___OVERLAPPED__'___
FLOATING_SAVE_AREA
_1__IID__„__DLGTEMPLATE_____PROPSHEETPAGEA__'__LPDLGTEMPLATEA__f
__PROPSHEETPAGEW
_9__sval__"__tagPOINT__—___CONTEXT__™__RPC_VERSION
_"__POINT__—__CONTEXT
_3__LPIID__‰__PEXCEPTION_RECORD
_9__SVAL__›__val_context
____HDWP__H__cState
_u__PUINT_____RPC_SS_THREAD_HANDLE__X__CALINFO_ENUMPROCW_____R
PC_AUTH_IDENTITY_HANDLE__6_
ENHMETARECORD&_
_SECURITY_CONTEXT_TRACKING_MODE__X__DATEFMT_ENUMPROCW__t_
PSID_NAME_USE___MIDICALLBACK@20"_!_SECURITY_DESCRIPTOR_CONTROL
____
       EXPR_EVAL__ž_LPFRHOOKPROC__X__CALINFO_ENUMPROCA__¡__PFNC
ALLBACK__X__DATEFMT_ENUMPROCA__¤_ICMENUMPROCW__©_
PQUERYHANDLER
_@_FLOAT__"__LPDWORD_____RPC_AUTHZ_HANDLE__uMMVERSION
____HRSRC___RPC_FORWARD_FUNCTION@20"_®__PTOP_LEVEL_EXCEPTION_F
ILTER
____HCONV

_BYTE____LONGLONG__a__LPPROPSHEETPAGEW__±_ICMENUMPROCA__!__LP
WSTR__"_ACCESS_MASK__"_COLORREF__Z__LPPROPSHEETPAGEA__´__PIMAG
E_TLS_CALLBACK_____HWAVEOUT__$_PRPC_RUNDOWN____HENHMETAFILE
_t__PBOOL__‡__RPC_SYNTAX_IDENTIFIER__X__DESKTOPENUMPROCW__"_u_lo
ng
_t__SPTR__!__LPOLESTR____NDR_CCONTEXT
_u_UINT__X__WNDENUMPROC__X__DESKTOPENUMPROCA__¶__WNDPROC__"_
HREFTYPE__#      DWORDLONG
_p__LPSTR__t
TOKEN_TYPE
_!__BSTR____
RPC_NS_HANDLE
____HDESK__"__PDWORD__°___EXCEPTION_POINTERS__!
CLIPFORMAT__!__LPWORD__t_TYPEKIND__X__CODEPAGE_ENUMPROCW
_p__HPSTR__d__LPFNPSPCALLBACKW__!_OLECHAR__X__ENUMRESLANGPROC
__t_VARKIND__X__CODEPAGE_ENUMPROCA__]__LPFNPSPCALLBACKA__ _
```

```
RPC_BUFPTR__X__WINSTAENUMPROCW__½__RPC_AUTH_KEY_RETRIEVAL_FN
__!_ LPCOLESTR____ SC_HANDLE
____HTASK__À__REGISTERWORDENUMPROCW__X__TIMEFMT_ENUMPROCW__
X__WINSTAENUMPROCA
_!_ATOM__!__LPCWSTR__X__DLGPROC__Ã__REGISTERWORDENUMPROCA____
_HOLEMENU
_t_BOOL___RPC_OBJECT_INQ_FN@12
_"_ULONG__X__TIMEFMT_ENUMPROCA__«__PEXCEPTION_POINTERS__Æ__tag
HELPINFO__t_CALLCONV_____LPMENUTEMPLATEA_____HGLOBAL_____LPVOI
D____
RPC_IF_HANDLE__X__HOOKPROC
_p__PSTR__ž_LPCFHOOKPROC__!_USHORT__7_
LPHMIDIOUT____HRESULT__
__PUCHAR____DISPID__È__PFULL_PTR_XLAT_TABLES__ž_LPCCHOOKPROC__
__LPBYTE
_X__PROC
_A_DATE__Ð_
ENHMFENUMPROC
____PSID
_t__LPINT
_Ñ__SNB
_p_CHAR__p__LPCSTR__Ó__SERVER_ROUTINE
___SCODE&_Ø__LPOVERLAPPED_COMPLETION_ROUTINE__Û_
      YIELDPROC__!_VARTYPE_____HWAVEIN__ß__IRpcChannelBuffer__ß__IRpc
ChannelBuffer__ž__LPPAGEPAINTHOOK__t
      OLERENDER__u__LPUINT__X_GOBJENUMPROC____
HMETAFILEPICT__Ñ__LPBSTR

_byte__@__PFLOAT__"__PSECURITY_INFORMATION__$__NDR_RUNDOWN_____
LPCVOID__"_SERVICE_STATUS_HANDLE__ž__LPPAGESETUPHOOK
_u_u_int__!_wint_t__â_
LPMMIOPROC____
STUB_THUNK
_!_WCHAR_____HGDIOBJ__X__ENUMRESNAMEPROC____HCONVLIST
____HSZ
_t_INT_____HDDEDATA__å__tagHANDLETABLE__X__FARPROC__X_
PROPENUMPROCW__"_MCIERROR__t__LPBOOL
____PLONG__u_WPARAM
____HMMIO__ž_
LPOFNHOOKPROC__ç__HPROPSHEETPAGE__QUERYHANDLER__X_
PROPENUMPROCA
____HKL_____HACCEL__"
RPCOLEDATAREP___VARIANT_BOOL_____HCURSOR__7__PHANDLE__é_
    _onexit_t__X_
FONTENUMPROCW__"_LCTYPE__î__RPC_DISPATCH_FUNCTION
_"_CALID___
```

```
RPC_STATUS_____HMIDIIN
____HDC__ó__MSGBOXCALLBACK__X_
FONTENUMPROCA_____GENERIC_UNBIND_ROUTINE_____HBITMAP____LPARA
M___
MENUTEMPLATEW__7__LPHMIDISTRM__t_TOKEN_INFORMATION_CLASS___D
RVCALLBACK@20__÷__RPC_MESSAGE___
MENUTEMPLATEA
____HMIDI__ú__PFNPROPSHEETCALLBACK_____PSECURITY_DESCRIPTOR
____HMENU_____HWINSTA___
LCSGAMUTMATCH___RPC_CLIENT_FREE@4__ý__PMIDL_STUB_DESC
__LONG____NDR_SCONTEXT____
ARRAY_INFO__p__va_list__7__LPHMIXER__X_      TIMERPROC__X_
DRAWSTATEPROC____    HINSTANCE__"_FOURCC__
__MIDL_STUB_MESSAGE__"_SECURITY_INFORMATION__X__ENUMRESTYPEP
ROC__å__HANDLETABLE__7_   LPHWAVEIN__
__PFORMAT_STRING__"_CALTYPE__7__LPHANDLE_____RPC_MGMT_AUTHORI
ZATION_FN__!_LANGID"_t_SECURITY_IMPERSONATION_LEVEL____fpos_t__u_S
OCKET___BOOLEAN__7__RPC_EP_INQ_HANDLE
____HWND___MMIOPROC@16_____LPHANDLER_FUNCTION__t_EXCEPTION_DIS
POSITION__tXLAT_SIDE
____HHOOK_____LPTHREAD_START_ROUTINE__7_
LPHWAVEOUT_____HPALETTE__X__PROPENUMPROCEXW_____HMODULE___W
AVECALLBACK@20___RPC_CLIENT_ALLOC@4_____HANDLE_____HMIXER
_!__PWORD__X__PROPENUMPROCEXA_____HCOLORSPACE__X__EDITWORDBR
EAKPROCA_____GENERIC_BINDING_ROUTINE__!_u_short__7_
LPHMIDIIN__X__LOCALE_ENUMPROCW
_"_DWORD__X__GRAYSTRINGPROC__t_DESCKIND
_7__PHKEY"_®__LPTOP_LEVEL_EXCEPTION_FILTER
_"_LCID_____LPFNADDPROPSHEETPAGE___u_char__X__LOCALE_ENUMPROCA
____HRGN__t_ACL_INFORMATION_CLASS__t_SYSKIND___TIMECALLBACK@20_
___
PACCESS_TOKEN_____SC_LOCK__X__LINEDDAPROC_____PHANDLER_ROUTIN
E"_!__PSECURITY_DESCRIPTOR_CONTROL__t_FUNCKIND___
FXPT2DOT30_____LPLONG
____HGLRC_____HMIDIOUT__t
INVOKEKIND____  HMIDISTRM
____HFONT
____HPEN
____HICON
____PVOID____
tagMETARECORD__"_REGSAM__t      OLEUPDATE__u_MCIDEVICEID__
__LPSERVICE_MAIN_FUNCTIONW
_!_WORD____        HMETAFILE__X_   ABORTPROC
____HDRVR__$__LPSERVICE_MAIN_FUNCTIONA__ž__LPPRINTHOOKPROC__u_si
ze_t___       LCSCSTYPE
___PBYTE____LRESULT__)_
```

```
MFENUMPROC
_é__FPTR
____HKEY
_ _UCHAR_____PTHREAD_START_ROUTINE__!_wctype_t____I_RPC_HANDLE
____HDROP____MEMBERID
___SHORT__!_wchar_t__u_MMRESULT_____XMIT_HELPER_ROUTINE_____RPC_BI
NDING_HANDLE
_"_HIMC_____I_RPC_MUTEX____
     HMIXEROBJ__ž__LPSETUPHOOKPROC__,__RPC_BLOCKING_FUNCTION__
___HBRUSH__X_
SENDASYNCPROC__!FILEOP_FLAGS__/__LPTIMECALLBACK_____handle_t_____H
LOCAL
_t_HFILE___ _‹___<_‹__J:\Vb60\Line21\funct.c_±_¦_»_Å_Ê_Ð_Þ___
_P_a_□_½_ù_________M_^_Ž_Ÿ_Í_û_7_A_F_L_{_È_!_K_U_Z_`_Ž_Û_ì___-
_▒_□_À_Ñ_Ö___#_S_d_i_□_š_Ÿ_¥_´_¾_Ã_Ï_Ó_à_í___U_q_{_€_‰_□_ž_®_¼_Í_Ò_à_
ñ_û__
     _____._5_:_A_N_\_v_„_ž_¬_Æ_Ð_Õ_Þ_å_ó_____"_'_5_F_P_U_^_e_s_ƒ_‘_¢_§_µ
_Æ_Ð_Õ_Þ_å_ó_______$_._3_9_C_M_W_\_e_l_ˆ_˜_¦_«_¹_Ã_È_Ñ_Ø_å_ò___!_+_5_F_p_
u_†_°_µ_¿_Ð_á_æ_÷___
___!_2_\_a_r_œ_¡_«_¼_Ï_Ò_ã_ô_2_[_}_‡__________ "#$%&-
./12345689@ABCDEFOPQSTUWYZ[]_abcefhiopqry|}~▒□„…Š□Ž□‘’”—
™š□ ¡¢£¤¥¦§¨©ª«¬-®±´µ¶¸¹»½¾ÀÁÄÇÈÉÊÌÎÐÑÓÔÚÝÞßàáâãäèìíîïõøùûüýþ___
_
____________________!_#_%_&_(_*_+_/_1_3_5_6_8_:_;_>_@_B_C_E_G_H_L_M_
N_P_Q___    _Debug/Parity.objF__<Microsoft (R) 32-bit C/C++ Optimizing Compiler
Version 10.00___
_I__parity__  _Debug/Tests.objF__<Microsoft (R) 32-bit C/C++ Optimizing Compiler
Version 10.002__Ü___`       _3___DefaultTest_____field__ _ch___p__mes___1_
     cStatePtr__.__T_._)u _3___NullTest_____field__ _ch___p__mes___1_
     cStatePtr__.__Ì_9_4£ _3___  F1XDSTest_____field__ _ch___p__mes___1_
     cStatePtr__2__H_F_AÜ     _3__XDSStartTest_____field__ _ch___p__mes___1_
     cStatePtr__2__Ä_I_D"
_3___XDSContTest_____field__ _ch___p__mes___1_    cStatePtr__.__<_/_*k
_3__
XDSEndTest_____field__ _ch___p__mes___1_    cStatePtr__2__,_=_8š
_3___ControlTest_____field__ _ch___p__mes___1_     cStatePtr__2__@_p   k×
_3__TextModeTest_____field__ _ch___p__mes___1_     cStatePtr
_üÿÿÿt_c__.__Ä_p   kG__3__
RollUpTest_____field__ _ch___p__mes___1_     cStatePtr
_üÿÿÿt_c__2__L_b   ]·_3___PaintOnTest_____field__ _ch___p__mes___1_   cStatePtr
_üÿÿÿt_c__.__Ð_b   ]__3__PopOnTest_____field__ _ch___p__mes___1_      cStatePtr
_üÿÿÿt_c__6__\_b   ]{_3___EndOfCaptionTest_____field__ _ch___p__mes___1_
     cStatePtr
_üÿÿÿt_c__.__à_N   ÌÝ_3___PACTest_____field__ _ch___p__mes___1_       cStatePtr
_üÿÿÿt_c__2__„_ˆ    ƒ+
_3___CtrlChangeTest_____field__ _ch___p__mes___1_   cStatePtr
```

```
_ôÿÿÿt_f__øÿÿÿt_ctrl
_üÿÿÿt_c__*____b    ]³
_3___CRTest_____field__ _ch___p__mes___1_    cStatePtr
_üÿÿÿt_c__.__^_\      W___3___URLTest_____field__ _ch___p__mes___1_     cStatePtr
_üÿÿÿt_c__.__$        V_     Q_q__3__
URLEndTest_____field__ _ch___p__mes___1_    cStatePtr
_ôÿÿÿt_l
_øÿÿÿt_f
_üÿÿÿp__t________XMIT_ROUTINE_QUINTUPLE_______GENERIC_BINDING_INFO
_____tagRPCOLEMESSAGE__&___MALLOC_FREE_STRUCT"_(___GENERIC_BINDI
NG_ROUTINE_PAIR__&__MALLOC_FREE_STRUCT__*___COMM_FAULT_OFFSET
S"_.___FULL_PTR_TO_REFID_ELEMENT"_(__GENERIC_BINDING_ROUTINE_PAIR
_____XMIT_ROUTINE_QUINTUPLE"_,__PFULL_PTR_TO_REFID_ELEMENT
_1___GUID__3__LPCLSID__6__tagENHMETARECORD
_1__GUID__L___MIDL_STUB_DESC__S___FULL_PTR_XLAT_TABLES______PROP
SHEETPAGEA__f___PROPSHEETPAGEW
_1__CLSID__7__XDSState__,__IRpcChannelBufferVtbl__„__DLGTEMPLATE__‡___RP
C_SYNTAX_IDENTIFIER__Œ___EXCEPTION_RECORD__Ž___OVERLAPPED__‘___
FLOATING_SAVE_AREA
_1__IID__„__DLGTEMPLATE_____PROPSHEETPAGEA__’__LPDLGTEMPLATEA__f
__PROPSHEETPAGEW
_9__sval__”__tagPOINT__—___CONTEXT__™__RPC_VERSION
_”__POINT__—__CONTEXT
_3__LPIID__‰__PEXCEPTION_RECORD
_9__SVAL__›__val_context
____HDWP__H__cState
_u__PUINT_____RPC_SS_THREAD_HANDLE__X__CALINFO_ENUMPROCW_____R
PC_AUTH_IDENTITY_HANDLE__6_
ENHMETARECORD&_
_SECURITY_CONTEXT_TRACKING_MODE__X__DATEFMT_ENUMPROCW__t_
PSID_NAME_USE___MIDICALLBACK@20"_!_SECURITY_DESCRIPTOR_CONTROL

____
      EXPR_EVAL__ž_LPFRHOOKPROC__X__CALINFO_ENUMPROCA__¡__PFNC
ALLBACK__X__DATEFMT_ENUMPROCA__¤_ICMENUMPROCW__©_
PQUERYHANDLER
_@_FLOAT__"__LPDWORD_____RPC_AUTHZ_HANDLE__uMMVERSION
____HRSRC___RPC_FORWARD_FUNCTION@20"_®__PTOP_LEVEL_EXCEPTION_F
ILTER
____HCONV
_
_BYTE____LONGLONG__a__LPPROPSHEETPAGEW__±_ICMENUMPROCA__!__LP
WSTR__"_ACCESS_MASK__"_COLORREF__Z__LPPROPSHEETPAGEA__´__PIMAG
E_TLS_CALLBACK_____HWAVEOUT__$_PRPC_RUNDOWN____HENHMETAFILE
_t__PBOOL__‡__RPC_SYNTAX_IDENTIFIER__X__DESKTOPENUMPROCW__"_u_lo
ng
_t__SPTR__!__LPOLESTR____NDR_CCONTEXT
_u_UINT__X__WNDENUMPROC__X__DESKTOPENUMPROCA__¶__WNDPROC__"_
HREFTYPE__#       DWORDLONG
_p__LPSTR__t
TOKEN_TYPE
_!__BSTR____
RPC_NS_HANDLE
____HDESK__"__PDWORD__°___EXCEPTION_POINTERS__!
CLIPFORMAT__!__LPWORD__t_TYPEKIND__X__CODEPAGE_ENUMPROCW
_p__HPSTR__d__LPFNPSPCALLBACKW__!_OLECHAR__X__ENUMRESLANGPROC
__t_VARKIND__X__CODEPAGE_ENUMPROCA__]__LPFNPSPCALLBACKA__ _
RPC_BUFPTR__X__WINSTAENUMPROCW__½__RPC_AUTH_KEY_RETRIEVAL_FN
__!_  LPCOLESTR____     SC_HANDLE
____HTASK__À__REGISTERWORDENUMPROCW__X__TIMEFMT_ENUMPROCW__
X__WINSTAENUMPROCA
_!_ATOM__!__LPCWSTR__X__DLGPROC__Ã__REGISTERWORDENUMPROCA____
_HOLEMENU
_t_BOOL___RPC_OBJECT_INQ_FN@12
_"_ULONG__X__TIMEFMT_ENUMPROCA__«__PEXCEPTION_POINTERS__Æ__tag
HELPINFO__t_CALLCONV_____LPMENUTEMPLATEA_____HGLOBAL_____LPVOI
D____
RPC_IF_HANDLE__X__HOOKPROC
_p__PSTR__ž_LPCFHOOKPROC__!_USHORT__7_
LPHMIDIOUT____HRESULT__
__PUCHAR____DISPID__È__PFULL_PTR_XLAT_TABLES__ž_LPCCHOOKPROC__
__LPBYTE
_X__PROC
_A_DATE__Đ_
ENHMFENUMPROC
____PSID
_t__LPINT
_Ñ__SNB
_p_CHAR__p__LPCSTR__Ó__SERVER_ROUTINE
___SCODE&_Ø__LPOVERLAPPED_COMPLETION_ROUTINE__Û_
        YIELDPROC__!_VARTYPE_____HWAVEIN__ß__IRpcChannelBuffer__ß__IRpc
ChannelBuffer__ž__LPPAGEPAINTHOOK__t
        OLERENDER__u__LPUINT__X_GOBJENUMPROC____
HMETAFILEPICT__Ñ__LPBSTR

_
_byte__@__PFLOAT__"__PSECURITY_INFORMATION__$__NDR_RUNDOWN_____
LPCVOID__"_SERVICE_STATUS_HANDLE__ž__LPPAGESETUPHOOK
_u_u_int__!_wint_t__â_
LPMMIOPROC____
STUB_THUNK
_!_WCHAR_____HGDIOBJ__X__ENUMRESNAMEPROC____HCONVLIST
____HSZ
_t_INT_____HDDEDATA__å__tagHANDLETABLE__X__FARPROC__X_
```

```
PROPENUMPROCW__"_MCIERROR__t__LPBOOL
____PLONG__u_WPARAM
____HMMIO__ž_
LPOFNHOOKPROC__ç__HPROPSHEETPAGE__QUERYHANDLER__X_
PROPENUMPROCA
____HKL_____HACCEL__"
RPCOLEDATAREP___VARIANT_BOOL_____HCURSOR__7__PHANDLE__é_
     _onexit_t__X_
FONTENUMPROCW__"_LCTYPE__î__RPC_DISPATCH_FUNCTION
_"_CALID___
RPC_STATUS_____HMIDIIN
____HDC__ó__MSGBOXCALLBACK__X_
FONTENUMPROCA_____GENERIC_UNBIND_ROUTINE_____HBITMAP____LPARA
M___
MENUTEMPLATEW__7__LPHMIDISTRM__t_TOKEN_INFORMATION_CLASS___D
RVCALLBACK@20__÷__RPC_MESSAGE___
MENUTEMPLATEA
____HMIDI__ú__PFNPROPSHEETCALLBACK_____PSECURITY_DESCRIPTOR
____HMENU_____HWINSTA___
LCSGAMUTMATCH___RPC_CLIENT_FREE@4__ý__PMIDL_STUB_DESC
___LONG____NDR_SCONTEXT____
ARRAY_INFO__p__va_list__7__LPHMIXER__X_      TIMERPROC__X_
DRAWSTATEPROC____   HINSTANCE__"_FOURCC__
___MIDL_STUB_MESSAGE__"_SECURITY_INFORMATION__X__ENUMRESTYPEP
ROC__å__HANDLETABLE__7_   LPHWAVEIN__
__PFORMAT_STRING__"_CALTYPE__7__LPHANDLE_____RPC_MGMT_AUTHORI
ZATION_FN__!_LANGID"_t_SECURITY_IMPERSONATION_LEVEL____fpos_t__u_S
OCKET__ _BOOLEAN__7__RPC_EP_INQ_HANDLE
____HWND___MMIOPROC@16_____LPHANDLER_FUNCTION__t_EXCEPTION_DIS
POSITION__t XLAT_SIDE
____HHOOK_____LPTHREAD_START_ROUTINE__7_
LPHWAVEOUT_____HPALETTE__X__PROPENUMPROCEXW_____HMODULE___W
AVECALLBACK@20___RPC_CLIENT_ALLOC@4_____HANDLE_____HMIXER
_!__PWORD__X__PROPENUMPROCEXA_____HCOLORSPACE__X__EDITWORDBR
EAKPROCA_____GENERIC_BINDING_ROUTINE__!_u_short__7_
LPHMIDIIN__X__LOCALE_ENUMPROCW
_"_DWORD__X__GRAYSTRINGPROC__t_DESCKIND
_7__PHKEY"_®__LPTOP_LEVEL_EXCEPTION_FILTER
_"_LCID_____LPFNADDPROPSHEETPAGE__ _u_char__X__LOCALE_ENUMPROCA
____HRGN__t_ACL_INFORMATION_CLASS__t_SYSKIND___TIMECALLBACK@20_

___
PACCESS_TOKEN_____SC_LOCK__X__LINEDDAPROC_____PHANDLER_ROUTIN
E"_!__PSECURITY_DESCRIPTOR_CONTROL__t_FUNCKIND___
FXPT2DOT30_____LPLONG
____HGLRC_____HMIDIOUT__t
INVOKEKIND____  HMIDISTRM
```

```
____HFONT
____HPEN
____HICON
____PVOID____
tagMETARECORD__"_REGSAM__t        OLEUPDATE__u_MCIDEVICEID__
__LPSERVICE_MAIN_FUNCTIONW
_!_WORD____           HMETAFILE__X_    ABORTPROC
____HDRVR__$__LPSERVICE_MAIN_FUNCTIONA__ž__LPPRINTHOOKPROC__u_si
ze_t___           LCSCSTYPE
_ __PBYTE____LRESULT__)_
MFENUMPROC
_é__FPTR
____HKEY
_ _UCHAR_____PTHREAD_START_ROUTINE__!_wctype_t____I_RPC_HANDLE
____HDROP____MEMBERID
___SHORT__!_wchar_t__u_MMRESULT_____XMIT_HELPER_ROUTINE_____RPC_BI
NDING_HANDLE
_"_HIMC_____I_RPC_MUTEX____
      HMIXEROBJ__ž__LPSETUPHOOKPROC__,__RPC_BLOCKING_FUNCTION__
___HBRUSH__X_
SENDASYNCPROC__!FILEOP_FLAGS__/__LPTIMECALLBACK_____handle_t_____H
LOCAL
_t_HFILE___`Æ___<`        Æ__J:\Vb60\Line21\Tests.c_u`     f     p     u     {
       ^      ’.     —     ž     £     ©     Á     Ë     Đ     ×     Ü     â
      _
_
_
_
"
(
P
Z
_
f
k
q
▒
‰
Ž
•
š

¼
Æ
Ë
Ò
```

```
×
à
í
__1_;_B_G_P_]_…_¡_«_²_·_À_Í_õ__
__"/Weov{„‘¹ÇÑØÝæó_
_
_
&
+
4
;
H
T
˜
¢
§
®
³
¼
É
ñ
ÿ
        _______+_?_[_e_l_q_z_□_”_¥_¯_Ñ_I_Á_9_±_»_Â___
______________ !$%&'(),-
./0156789:=@ACDFGJMNPQSTWZ[]^`adghjkmnqtuwxz{~□,„…‡ˆ‹Ž□□‘’“”•˜›œžŸ¡¢¥¨©
«¬®¯²¶·¸¹º¼½¾¿ÀÂÃ__      _Debug/XDSMessages.objF__<Microsoft (R) 32-bit C/C++
Optimizing Compiler Version 10.00*__¼é_äP__K___Def___p__mes__1_
        cStatePtr__*__P_"_   __9__K___Vcp___p__mes__1_
        cStatePtr__ðÿÿÿp_b2__ôÿÿÿp_b1__øÿÿÿp_type__üÿÿÿp_flags__*__°_ú
        õ[__K___Typ___p__mes__1_        cStatePtr
_üÿÿÿt_i___________XDSMes____:_4__mpaaRatings___8:_4__tvpgRatings___X:_L__flag
mask___`:_4__types______XMIT_ROUTINE_QUINTUPLE_______GENERIC_BINDING
_INFO_____tagRPCOLEMESSAGE__&___MALLOC_FREE_STRUCT"_(___GENERIC_
BINDING_ROUTINE_PAIR__&__MALLOC_FREE_STRUCT__*___COMM_FAULT_O
FFSETS"_.___FULL_PTR_TO_REFID_ELEMENT"_(__GENERIC_BINDING_ROUTINE
_PAIR_____XMIT_ROUTINE_QUINTUPLE"_,__PFULL_PTR_TO_REFID_ELEMENT
_1___GUID__3__LPCLSID__6__tagENHMETARECORD
_1__GUID__L___MIDL_STUB_DESC__S___FULL_PTR_XLAT_TABLES______PROP
SHEETPAGEA__f___PROPSHEETPAGEW
_1__CLSID__7__XDSState__,__IRpcChannelBufferVtbl__„__DLGTEMPLATE__‡___RP
C_SYNTAX_IDENTIFIER__Œ___EXCEPTION_RECORD__Ž___OVERLAPPED__‘___
FLOATING_SAVE_AREA
_1__IID__„__DLGTEMPLATE_____PROPSHEETPAGEA__’__LPDLGTEMPLATEA__f
__PROPSHEETPAGEW
_9__sval__”__tagPOINT__—___CONTEXT__™__RPC_VERSION
_”__POINT__—__CONTEXT
_3__LPIID__‰__PEXCEPTION_RECORD
_9__SVAL__›__val_context
____HDWP__H__cState
_u__PUINT_____RPC_SS_THREAD_HANDLE__X__CALINFO_ENUMPROCW_____R
PC_AUTH_IDENTITY_HANDLE__6_
ENHMETARECORD&_
_SECURITY_CONTEXT_TRACKING_MODE__X__DATEFMT_ENUMPROCW__t_
PSID_NAME_USE___MIDICALLBACK@20"_!_SECURITY_DESCRIPTOR_CONTROL

____
      EXPR_EVAL__ž_LPFRHOOKPROC__X__CALINFO_ENUMPROCA__¡__PFNC
ALLBACK__X__DATEFMT_ENUMPROCA__¤_ICMENUMPROCW__©_
PQUERYHANDLER
_@_FLOAT__"__LPDWORD_____RPC_AUTHZ_HANDLE__uMMVERSION
____HRSRC___RPC_FORWARD_FUNCTION@20"_®__PTOP_LEVEL_EXCEPTION_F
ILTER
____HCONV

_
_BYTE____LONGLONG__a__LPPROPSHEETPAGEW__±_ICMENUMPROCA__!__LP
WSTR__"_ACCESS_MASK__"_COLORREF__Z__LPPROPSHEETPAGEA__´__PIMAG
E_TLS_CALLBACK_____HWAVEOUT__$_PRPC_RUNDOWN____HENHMETAFILE
_t__PBOOL__‡__RPC_SYNTAX_IDENTIFIER__X__DESKTOPENUMPROCW__"_u_lo
ng
_t__SPTR__!__LPOLESTR____NDR_CCONTEXT
_u_UINT__X__WNDENUMPROC__X__DESKTOPENUMPROCA__¶__WNDPROC__"_
HREFTYPE__#      DWORDLONG
_p__LPSTR__t
TOKEN_TYPE
_!__BSTR____
RPC_NS_HANDLE
____HDESK__"__PDWORD__°___EXCEPTION_POINTERS__!
CLIPFORMAT__!__LPWORD__t_TYPEKIND__X__CODEPAGE_ENUMPROCW
_p__HPSTR__d__LPFNPSPCALLBACKW__!_OLECHAR__X__ENUMRESLANGPROC
__t_VARKIND__X__CODEPAGE_ENUMPROCA__]__LPFNPSPCALLBACKA___
RPC_BUFPTR__X__WINSTAENUMPROCW__½__RPC_AUTH_KEY_RETRIEVAL_FN
__!_  LPCOLESTR____   SC_HANDLE
____HTASK__À__REGISTERWORDENUMPROCW__X__TIMEFMT_ENUMPROCW__
X__WINSTAENUMPROCA
_!_ATOM__!__LPCWSTR__X__DLGPROC__Ã__REGISTERWORDENUMPROCA____
_HOLEMENU
_t_BOOL___RPC_OBJECT_INQ_FN@12
_"_ULONG__X__TIMEFMT_ENUMPROCA__«__PEXCEPTION_POINTERS__Æ__tag
HELPINFO__t_CALLCONV_____LPMENUTEMPLATEA_____HGLOBAL_____LPVOI
D____
RPC_IF_HANDLE__X__HOOKPROC
_p__PSTR__ž_LPCFHOOKPROC__!_USHORT__7_
```

```
LPHMIDIOUT____HRESULT__
__PUCHAR____DISPID__È__PFULL_PTR_XLAT_TABLES__ž_LPCCHOOKPROC__
__LPBYTE
_X__PROC
_A_DATE__Đ_
ENHMFENUMPROC
____PSID
_t__LPINT
_Ñ__SNB
_p_CHAR__p__LPCSTR__Ó__SERVER_ROUTINE
___SCODE&_Ø__LPOVERLAPPED_COMPLETION_ROUTINE__Û_
      YIELDPROC__!_VARTYPE_____HWAVEIN__ß__IRpcChannelBuffer__ß__IRpc
ChannelBuffer__ž__LPPAGEPAINTHOOK__t
      OLERENDER__u__LPUINT__X_GOBJENUMPROC____
HMETAFILEPICT__Ñ__LPBSTR

_
_byte__@__PFLOAT__"__PSECURITY_INFORMATION__$__NDR_RUNDOWN_____
LPCVOID__"_SERVICE_STATUS_HANDLE__ž__LPPAGESETUPHOOK
_u_u_int__!_wint_t__â_
LPMMIOPROC____
STUB_THUNK
_!_WCHAR_____HGDIOBJ__X__ENUMRESNAMEPROC____HCONVLIST
____HSZ
_t_INT_____HDDEDATA__å__tagHANDLETABLE__X__FARPROC__X_
PROPENUMPROCW__"_MCIERROR__t__LPBOOL
____PLONG__u_WPARAM
____HMMIO__ž_
LPOFNHOOKPROC__ç__HPROPSHEETPAGE__QUERYHANDLER__X_
PROPENUMPROCA
____HKL_____HACCEL__"
RPCOLEDATAREP___VARIANT_BOOL_____HCURSOR__7__PHANDLE__é_
      _onexit_t__X_
FONTENUMPROCW__"_LCTYPE__î__RPC_DISPATCH_FUNCTION
_"_CALID___
RPC_STATUS_____HMIDIIN
____HDC__ó__MSGBOXCALLBACK__X_
FONTENUMPROCA_____GENERIC_UNBIND_ROUTINE_____HBITMAP____LPARA
M___
MENUTEMPLATEW__7__LPHMIDISTRM__t_TOKEN_INFORMATION_CLASS___D
RVCALLBACK@20__÷__RPC_MESSAGE___
MENUTEMPLATEA
____HMIDI__ú__PFNPROPSHEETCALLBACK_____PSECURITY_DESCRIPTOR
____HMENU_____HWINSTA___
LCSGAMUTMATCH___RPC_CLIENT_FREE@4__ý__PMIDL_STUB_DESC
___LONG____NDR_SCONTEXT____
ARRAY_INFO__p__va_list__7__LPHMIXER__X_      TIMERPROC__X_
```

```
DRAWSTATEPROC____    HINSTANCE__"_FOURCC__
___MIDL_STUB_MESSAGE__"_SECURITY_INFORMATION__X__ENUMRESTYPEP
ROC__å__HANDLETABLE__7_   LPHWAVEIN__
__PFORMAT_STRING__"_CALTYPE__7__LPHANDLE_____RPC_MGMT_AUTHORI
ZATION_FN__!_LANGID"_t_SECURITY_IMPERSONATION_LEVEL____fpos_t__u_S
OCKET___BOOLEAN__7__RPC_EP_INQ_HANDLE
____HWND___MMIOPROC@16_____LPHANDLER_FUNCTION__t_EXCEPTION_DIS
POSITION__t XLAT_SIDE
____HHOOK_____LPTHREAD_START_ROUTINE__7_
LPHWAVEOUT_____HPALETTE__X__PROPENUMPROCEXW_____HMODULE___W
AVECALLBACK@20___RPC_CLIENT_ALLOC@4_____HANDLE_____HMIXER
_!__PWORD__X__PROPENUMPROCEXA_____HCOLORSPACE__X__EDITWORDBR
EAKPROCA_____GENERIC_BINDING_ROUTINE__!_u_short__7_
LPHMIDIIN__X__LOCALE_ENUMPROCW
_"_DWORD__X__GRAYSTRINGPROC__t_DESCKIND
_7__PHKEY"_®__LPTOP_LEVEL_EXCEPTION_FILTER
_"_LCID_____LPFNADDPROPSHEETPAGE___u_char__X__LOCALE_ENUMPROCA
____HRGN__t_ACL_INFORMATION_CLASS__t_SYSKIND___TIMECALLBACK@20_
___
PACCESS_TOKEN_____SC_LOCK__X__LINEDDAPROC_____PHANDLER_ROUTIN
E"_!__PSECURITY_DESCRIPTOR_CONTROL__t_FUNCKIND___
FXPT2DOT30_____LPLONG
____HGLRC_____HMIDIOUT__t
INVOKEKIND____  HMIDISTRM
____HFONT
____HPEN
____HICON
____PVOID____
tagMETARECORD__"_REGSAM__t      OLEUPDATE__u_MCIDEVICEID__
__LPSERVICE_MAIN_FUNCTIONW
_!_WORD____        HMETAFILE__X_  ABORTPROC
____HDRVR__$__LPSERVICE_MAIN_FUNCTIONA__ž__LPPRINTHOOKPROC__u_si
ze_t___      LCSCSTYPE
___PBYTE____LRESULT__)_
MFENUMPROC
_é__FPTR
____HKEY
___UCHAR_____PTHREAD_START_ROUTINE__!_wctype_t____I_RPC_HANDLE
____HDROP____MEMBERID
___SHORT__!_wchar_t__u_MMRESULT_____XMIT_HELPER_ROUTINE_____RPC_BI
NDING_HANDLE
_"_HIMC_____I_RPC_MUTEX____
        HMIXEROBJ__ž__LPSETUPHOOKPROC__,__RPC_BLOCKING_FUNCTION__
___HBRUSH__X_
SENDASYNCPROC__!FILEOP_FLAGS__/__LPTIMECALLBACK_____handle_t_____H
LOCAL
```

```
_t_HFILE___P_T___DP_T__J:\Vb60\Line21\XDSMessages.c_-P_V_g_—
_¨_ÿ_4_9_B_S_ƒ_°_Ý_é_____._@_Q_1_x_ˆ_˜_¯_»_Ì_Ø_é_õ_____#_/_@_E_V_[_d_u_¥_¶
_ø_:_K_P_______________"#%'()*+,-./01234568:<?BCDEGHIJ__
        _Debug/L21Dec.objF__<Microsoft (R) 32-bit C/C++ Optimizing Compiler Version
10.00.__ä___□__M___DllMain______hInst"_"_ul_reason_being_called_____
lpReserved__2__d_____§__O__Line21Decode_____field__ _ch___p__mes
_øÿÿÿt_j
_üÿÿÿt_i__2__à_Ê    ÅÁ__P__stateMachine_____field__
_ch___p__mes__üÿÿÿE__cstate__.__4____‹__R__
ShiftRight___"_value____shift__.__ˆ____¥__R__  ShiftLeft___"_value____shift_____0
_S_    DataState___°□_H__CCstate___\l_1_
        cStatePtr______XMIT_ROUTINE_QUINTUPLE_______GENERIC_BINDING_IN
FO_____tagRPCOLEMESSAGE__&___MALLOC_FREE_STRUCT"_(___GENERIC_BIN
DING_ROUTINE_PAIR__&__MALLOC_FREE_STRUCT__*___COMM_FAULT_OFFS
ETS"_.___FULL_PTR_TO_REFID_ELEMENT"_(__GENERIC_BINDING_ROUTINE_P
AIR_____XMIT_ROUTINE_QUINTUPLE"_,__PFULL_PTR_TO_REFID_ELEMENT
_1___GUID__3__LPCLSID__6__tagENHMETARECORD
_1__GUID__L___MIDL_STUB_DESC__S___FULL_PTR_XLAT_TABLES______PROP
SHEETPAGEA__f___PROPSHEETPAGEW
_1__CLSID__7__XDSState__,__IRpcChannelBufferVtbl__„__DLGTEMPLATE__‡___RP
C_SYNTAX_IDENTIFIER__Œ___EXCEPTION_RECORD__Ž___OVERLAPPED__‘___
FLOATING_SAVE_AREA
_1__IID__„__DLGTEMPLATE_____PROPSHEETPAGEA__’__LPDLGTEMPLATEA__f
__PROPSHEETPAGEW__”__tagPOINT__—___CONTEXT__™__RPC_VERSION
_”__POINT__—__CONTEXT
_3__LPIID__‰__PEXCEPTION_RECORD__›__val_context
____HDWP__H__cState
_u__PUINT_____RPC_SS_THREAD_HANDLE__X__CALINFO_ENUMPROCW_____R
PC_AUTH_IDENTITY_HANDLE__6_
ENHMETARECORD&_
_SECURITY_CONTEXT_TRACKING_MODE__X__DATEFMT_ENUMPROCW__t_
PSID_NAME_USE___MIDICALLBACK@20"_!_SECURITY_DESCRIPTOR_CONTROL

____    EXPR_EVAL__ž_LPFRHOOKPROC__X__CALINFO_ENUMPROCA__¡__PFNC
ALLBACK__X__DATEFMT_ENUMPROCA__¤_ICMENUMPROCW__©_
PQUERYHANDLER
_@_FLOAT__"__LPDWORD_____RPC_AUTHZ_HANDLE__uMMVERSION
____HRSRC___RPC_FORWARD_FUNCTION@20"_®__PTOP_LEVEL_EXCEPTION_F
ILTER
____HCONV

_BYTE____LONGLONG__a__LPPROPSHEETPAGEW__±_ICMENUMPROCA__!__LP
WSTR__"_ACCESS_MASK__"_COLORREF__Z__LPPROPSHEETPAGEA__´__PIMAG
E_TLS_CALLBACK_____HWAVEOUT__$_PRPC_RUNDOWN____HENHMETAFILE
_t__PBOOL__‡__RPC_SYNTAX_IDENTIFIER__X__DESKTOPENUMPROCW__"_u_lo
ng
_t__SPTR__!__LPOLESTR____NDR_CCONTEXT
_u_UINT__X__WNDENUMPROC__X__DESKTOPENUMPROCA__¶__WNDPROC__"_
HREFTYPE__#      DWORDLONG
_p__LPSTR__t
TOKEN_TYPE
_!__BSTR____
RPC_NS_HANDLE
____HDESK__"__PDWORD__°___EXCEPTION_POINTERS__!
CLIPFORMAT__!__LPWORD__t_TYPEKIND__X__CODEPAGE_ENUMPROCW
_p__HPSTR__d__LPFNPSPCALLBACKW__!_OLECHAR__X__ENUMRESLANGPROC
__t_VARKIND__X__CODEPAGE_ENUMPROCA__]__LPFNPSPCALLBACKA__ _
RPC_BUFPTR__X__WINSTAENUMPROCW__½__RPC_AUTH_KEY_RETRIEVAL_FN
__!_  LPCOLESTR____     SC_HANDLE
____HTASK__À__REGISTERWORDENUMPROCW__X__TIMEFMT_ENUMPROCW__
X__WINSTAENUMPROCA
_!_ATOM__!__LPCWSTR__X__DLGPROC__Ã__REGISTERWORDENUMPROCA____
_HOLEMENU
_t_BOOL___RPC_OBJECT_INQ_FN@12
_"_ULONG__X__TIMEFMT_ENUMPROCA__«__PEXCEPTION_POINTERS__Æ__tag
HELPINFO__t_CALLCONV_____LPMENUTEMPLATEA_____HGLOBAL_____LPVOI
D____
RPC_IF_HANDLE__X__HOOKPROC
_p__PSTR__ž_LPCFHOOKPROC__!_USHORT__7_
LPHMIDIOUT____HRESULT__
__PUCHAR____DISPID__È__PFULL_PTR_XLAT_TABLES__ž_LPCCHOOKPROC__
__LPBYTE
_X__PROC
_A_DATE__Ð_
ENHMFENUMPROC
____PSID
_t__LPINT
_Ñ__SNB
_p_CHAR__p__LPCSTR__Ó__SERVER_ROUTINE
___SCODE&_Ø__LPOVERLAPPED_COMPLETION_ROUTINE__Û_
      YIELDPROC__!_VARTYPE_____HWAVEIN__ß__IRpcChannelBuffer__ß__IRpc
ChannelBuffer__ž__LPPAGEPAINTHOOK__t
      OLERENDER__u__LPUINT__X_GOBJENUMPROC____
HMETAFILEPICT__Ñ__LPBSTR
_
_byte__@__PFLOAT__"__PSECURITY_INFORMATION__$__NDR_RUNDOWN_____
LPCVOID__"_SERVICE_STATUS_HANDLE__ž__LPPAGESETUPHOOK
_u_u_int__!_wint_t__â_
LPMMIOPROC____
STUB_THUNK
_9__sval
_!_WCHAR_____HGDIOBJ__X__ENUMRESNAMEPROC____HCONVLIST
```

```
___HSZ
_t_INT_____HDDEDATA__å__tagHANDLETABLE__X__FARPROC__X_
PROPENUMPROCW__"_MCIERROR__t__LPBOOL
___PLONG__u_WPARAM
___HMMIO__ž_
LPOFNHOOKPROC__ç__HPROPSHEETPAGE__QUERYHANDLER__X_
PROPENUMPROCA
___HKL_____HACCEL__"
RPCOLEDATAREP___VARIANT_BOOL_____HCURSOR__7__PHANDLE__é_
        _onexit_t__X_
FONTENUMPROCW__"_LCTYPE__î__RPC_DISPATCH_FUNCTION
_"_CALID___
RPC_STATUS_____HMIDIIN
___HDC__ó__MSGBOXCALLBACK__X_
FONTENUMPROCA_____GENERIC_UNBIND_ROUTINE_____HBITMAP____LPARA
M___
MENUTEMPLATEW__7__LPHMIDISTRM__t_TOKEN_INFORMATION_CLASS___D
RVCALLBACK@20__÷__RPC_MESSAGE___
MENUTEMPLATEA
___HMIDI__ú__PFNPROPSHEETCALLBACK_____PSECURITY_DESCRIPTOR
___HMENU_____HWINSTA___
LCSGAMUTMATCH___RPC_CLIENT_FREE@4__ý__PMIDL_STUB_DESC
___LONG____NDR_SCONTEXT____
ARRAY_INFO__p__va_list__7__LPHMIXER__X_      TIMERPROC__X_
DRAWSTATEPROC____   HINSTANCE__"_FOURCC__
__MIDL_STUB_MESSAGE__"_SECURITY_INFORMATION__X__ENUMRESTYPEP
ROC__å__HANDLETABLE__7_   LPHWAVEIN__
_PFORMAT_STRING__"_CALTYPE__7__LPHANDLE_____RPC_MGMT_AUTHORI
ZATION_FN__!_LANGID"_t_SECURITY_IMPERSONATION_LEVEL____fpos_t__u_S
OCKET____BOOLEAN__7__RPC_EP_INQ_HANDLE
___HWND___MMIOPROC@16_____LPHANDLER_FUNCTION__t_EXCEPTION_DIS
POSITION__tXLAT_SIDE
___HHOOK_____LPTHREAD_START_ROUTINE__7_
LPHWAVEOUT_____HPALETTE__X__PROPENUMPROCEXW_____HMODULE___W
AVECALLBACK@20___RPC_CLIENT_ALLOC@4_____HANDLE_____HMIXER
_!__PWORD__X__PROPENUMPROCEXA_____HCOLORSPACE__X__EDITWORDBR
EAKPROCA_____GENERIC_BINDING_ROUTINE__!_u_short__7_
LPHMIDIIN__X__LOCALE_ENUMPROCW
_"_DWORD__X__GRAYSTRINGPROC__t_DESCKIND
_7__PHKEY"_®__LPTOP_LEVEL_EXCEPTION_FILTER
_"_LCID_____LPFNADDPROPSHEETPAGE___u_char__X__LOCALE_ENUMPROCA
___HRGN__t_ACL_INFORMATION_CLASS__t_SYSKIND___TIMECALLBACK@20_
___

PACCESS_TOKEN_____SC_LOCK__X__LINEDDAPROC_____PHANDLER_ROUTIN
E"_!__PSECURITY_DESCRIPTOR_CONTROL__t_FUNCKIND___
FXPT2DOT30_____LPLONG
```

```
___HGLRC____HMIDIOUT__t
INVOKEKIND___ HMIDISTRM
___HFONT
___HPEN
___HICON
___PVOID___
tagMETARECORD__"_REGSAM__t      OLEUPDATE__u_MCIDEVICEID__
_LPSERVICE_MAIN_FUNCTIONW
_!_WORD___          HMETAFILE__X_   ABORTPROC
___HDRVR__$__LPSERVICE_MAIN_FUNCTIONA__ž__LPPRINTHOOKPROC__u_si
ze_t__        LCSCSTYPE
_ _PBYTE___LRESULT__)_
MFENUMPROC
_é__FPTR
___HKEY
_ _UCHAR____PTHREAD_START_ROUTINE__!_wctype_t____I_RPC_HANDLE
___HDROP___MEMBERID
___SHORT__!_wchar_t__u_MMRESULT_____XMIT_HELPER_ROUTINE_____RPC_BI
NDING_HANDLE
_"_HIMC_____I_RPC_MUTEX____
     HMIXEROBJ__ž__LPSETUPHOOKPROC__,__RPC_BLOCKING_FUNCTION__
___HBRUSH__X_
SENDASYNCPROC__!FILEOP_FLAGS__/__LPTIMECALLBACK_____handle_t_____H
LOCAL
_t_HFILE___□_¾___<□_¾__J:\Vb60\Line21\L21Dec.c_*□_–
_ _§_° °_Ä_Ó_â_î_ú_____+_G_q_v_□_™_Ÿ_³_°_Á_Ê_Ý_____=_B_X_]_b_x_}_□_†_‹_‘
_ž_¥_«_,_________ !"#%&')+,-/04@CEGHIJLNOPRSVWX[\]__
     _Debug/stable.objF__<Microsoft (R) 32-bit C/C++ Optimizing Compiler Version
10.00___0
_T_   DataState___˜
_U_ControlState___□__V_
XDSStartState___°__V_XDSContState___Ð__W_XDSDataState___
_X__XDSEndState___0_Y_URLDataState______XMIT_ROUTINE_QUINTUPLE______
GENERIC_BINDING_INFO_____tagRPCOLEMESSAGE__&___MALLOC_FREE_STRU
CT"_(___GENERIC_BINDING_ROUTINE_PAIR__&__MALLOC_FREE_STRUCT__*__
_COMM_FAULT_OFFSETS"_.___FULL_PTR_TO_REFID_ELEMENT"_(__GENERIC_
BINDING_ROUTINE_PAIR_____XMIT_ROUTINE_QUINTUPLE"_,__PFULL_PTR_TO
_REFID_ELEMENT
_1___GUID__3__LPCLSID__6__tagENHMETARECORD
_1__GUID__L___MIDL_STUB_DESC__S___FULL_PTR_XLAT_TABLES______PROP
SHEETPAGEA__f__PROPSHEETPAGEW
_1__CLSID__,__IRpcChannelBufferVtbl__„__DLGTEMPLATE__‡___RPC_SYNTAX_ID
ENTIFIER__Œ___EXCEPTION_RECORD__Ž___OVERLAPPED__‘___FLOATING_SA
VE_AREA
_1__IID__„__DLGTEMPLATE_____PROPSHEETPAGEA__’__LPDLGTEMPLATEA__f
_PROPSHEETPAGEW__”__tagPOINT__—___CONTEXT__™__RPC_VERSION
```

```
_"__POINT__—__CONTEXT
_3__LPIID__‰__PEXCEPTION_RECORD__›__val_context
____HDWP
_u__PUINT_____RPC_SS_THREAD_HANDLE__X__CALINFO_ENUMPROCW_____R
PC_AUTH_IDENTITY_HANDLE__6_
ENHMETARECORD&_
_SECURITY_CONTEXT_TRACKING_MODE__X__DATEFMT_ENUMPROCW__t_
PSID_NAME_USE___MIDICALLBACK@20"_!_SECURITY_DESCRIPTOR_CONTROL

_____
      EXPR_EVAL__ž_LPFRHOOKPROC__X__CALINFO_ENUMPROCA__¡__PFNC
ALLBACK__X__DATEFMT_ENUMPROCA__¤_ICMENUMPROCW__©_
PQUERYHANDLER
_@_FLOAT__"__LPDWORD_____RPC_AUTHZ_HANDLE__uMMVERSION
____HRSRC___RPC_FORWARD_FUNCTION@20"_®__PTOP_LEVEL_EXCEPTION_F
ILTER
____HCONV

_
_BYTE____LONGLONG__a__LPPROPSHEETPAGEW__±_ICMENUMPROCA__!__LP
WSTR__"_ACCESS_MASK__"_COLORREF__Z__LPPROPSHEETPAGEA__´__PIMAG
E_TLS_CALLBACK_____HWAVEOUT__$_PRPC_RUNDOWN____HENHMETAFILE
_t__PBOOL__‡__RPC_SYNTAX_IDENTIFIER__X__DESKTOPENUMPROCW__"_u_lo
ng
_t__SPTR__!__LPOLESTR____NDR_CCONTEXT
_u_UINT__X__WNDENUMPROC__X__DESKTOPENUMPROCA__¶__WNDPROC__"_
HREFTYPE__#      DWORDLONG
_p__LPSTR__t
TOKEN_TYPE
_!__BSTR____
RPC_NS_HANDLE
____HDESK__"__PDWORD__°___EXCEPTION_POINTERS__!
CLIPFORMAT__!__LPWORD__t_TYPEKIND__X__CODEPAGE_ENUMPROCW
_p__HPSTR__d__LPFNPSPCALLBACKW__!_OLECHAR__X__ENUMRESLANGPROC
__t_VARKIND__X__CODEPAGE_ENUMPROCA__]__LPFNPSPCALLBACKA__ _
RPC_BUFPTR__X__WINSTAENUMPROCW__½__RPC_AUTH_KEY_RETRIEVAL_FN
__!_   LPCOLESTR____     SC_HANDLE
____HTASK__À__REGISTERWORDENUMPROCW__X__TIMEFMT_ENUMPROCW__
X__WINSTAENUMPROCA
_!_ATOM__!__LPCWSTR__X__DLGPROC__Ã__REGISTERWORDENUMPROCA____
_HOLEMENU
_t_BOOL___RPC_OBJECT_INQ_FN@12
_"_ULONG__X__TIMEFMT_ENUMPROCA__«__PEXCEPTION_POINTERS__Æ__tag
HELPINFO__t_CALLCONV_____LPMENUTEMPLATEA_____HGLOBAL_____LPVOI
D____
RPC_IF_HANDLE__X__HOOKPROC
_p__PSTR__ž_LPCFHOOKPROC__!_USHORT__7_
```

```
LPHMIDIOUT____HRESULT__
__PUCHAR____DISPID__È__PFULL_PTR_XLAT_TABLES__ž_LPCCHOOKPROC__
__LPBYTE
_X__PROC
_A_DATE__Đ_
ENHMFENUMPROC
____PSID
_t__LPINT
_Ñ__SNB
_p_CHAR__p__LPCSTR__Ó__SERVER_ROUTINE
___SCODE&_Ø__LPOVERLAPPED_COMPLETION_ROUTINE__Û_
      YIELDPROC__!_VARTYPE_____HWAVEIN__ß__IRpcChannelBuffer__ß__IRpc
ChannelBuffer__ž__LPPAGEPAINTHOOK__t
      OLERENDER__u__LPUINT__X_GOBJENUMPROC____
HMETAFILEPICT__Ñ__LPBSTR

_
_byte__@__PFLOAT__"__PSECURITY_INFORMATION__$__NDR_RUNDOWN_____
LPCVOID__"_SERVICE_STATUS_HANDLE__ž__LPPAGESETUPHOOK
_u_u_int__!_wint_t__â_
LPMMIOPROC____
STUB_THUNK
_9__sval
_!_WCHAR_____HGDIOBJ__X__ENUMRESNAMEPROC____HCONVLIST
____HSZ
_t_INT_____HDDEDATA__å__tagHANDLETABLE__X__FARPROC__X_
PROPENUMPROCW__"_MCIERROR__t__L_€_€_____

_________________________________ (())**,,--
..//1122334455668899::;;<<==>>@@AABBCCDDEEFFGGHHIIJJKKLLMMNNOOPPQQ
RRSSTTUUVVWWXXYYZZ[[\\]]^^``ddeehhiikkllmmooqqssttwwxx{{||}}▒▒€€□□ƒƒ„„†
†^^%‰%‰ŠŠ‹‹ŒŒ□□ŽŽ□□’’””••——
™™©©ªª«««®®°°±±²²³³´´µµ¶¶·ºº»»¼¼ÀÀÁÁÂÂÃÃÆÆÍÍÏÏÒÒÓÓÔÔÕÕÖÖ××ØØÙÙÚÚ
ååççééêêëëîîïïððññ˜<€_@€8:€lB€,7€ì<€\<€xB€¸5€h3€¬9€,@€x4€@=€H5€Ì3€_3€Ä>€ÀA€8
@€ì>€@€„@€4€ðA€:€Ü@€ @€Ä;€$:€`6€Ì@€¨?€è6€ì7€ð2€hA€Ä<€ì=€´A€_A€„?€˜A€_3€
XA€p?€¼9€$9€HA€”>€„>€@7€à?€ð?€T@€¼@€8A€H;€È2€L6€P7€°8€X3€_@€ð3€Ü4€t5
€ø:€Ô>€$>€ø9€ÌC€(A€|3€ 4€t8€9€45€□3€D@€¤7€PC€¼B€¸3€´6€X=€`5€´>€´4€„<€¤>€_?
€Ü3€ì;€d7€¨=€¬@€□7€ˆ5€t@€\;€p;€Đ9€H<€d@€„;€B€ 5€¨:€p6€<€_;€T9€üB€<9€äA€àC€
_6€l9€ä9€Ø;€Ü8€°<€|C€°;€_4€h:€$C€_<€èB€_6€l=€€=€€9€0;€”=€7€È4€ô8€¬B€˜9€È6€t>
€œ8€È5€ˆ6€_?€¤A€_=€D>€à:€4>€A€d>€XB€X?€T>€ˆA€_C€_7€<B€¤3€¼C€À=€˜;€8C€l
C€Œ4€p<€,B€D8€_8€xA€,8€ü@€\8€ì@€P:€x7€À:€_5€8?€œ6€(?€Đ?€>€□C€0<€=€P4€Ü5
€ĐA€$4€ÔB€¸?€D3€_5€,3€Ü2€ C€_B€ŒB€Ä8€Ø=€|:€¬C€ˆ8€ð5€□:€□@€<4€È7€_ë__aut
odr__breake___case_R__char1“_consthk__continuei9__defaultkÿ__do_X_doublemQ__else
_,__enum
È__extern_°__floatq’_fors___gotot
__if_À_int_0_long____register▒ù__return_ÿ__short_
```

```
_signed∫m_sizeof     __static_T__struct†ø__switch_Í_typedef
___union_-__unsigned_›__void2:__volatile%oy__while_‘___inline(ó____cdecl._____based,,
I____stdcallÂ‹____declspec±Lasm²&_catch___classðÅ_const_cast□^deleteñÏ_dynamic_cas
t_çfriend_ñ_€inlineŽïnew™¢_operator‹I_privateŒ©_protected□»_publicéûreinterpret_castÏ
__static_cast³Étemplate’¹_this′&throwµ…_try{ô_typeid_&_virtualî—
_namespaceïH_using__›____single_inheritance__û____multiple_inheritance_______virtual_
inheritance__$____novtordisp__;____resume__!____nounwind__~____syscall__r____near_
_Á____far_______far16__ý____huge__-____fortran__d4€
___export_ô__interrupt_s___loadds_a___saveregs_Å___segment_²___segname_Ó___self__
ˆ___thiscall(ú_cdecl__Y_far__`__fortran__r__huge__ç__near__4__pascal__Ô___pascal©í_
__tryª____except«“____finally@___leave-
′____asm)Æ____fastcall__Œ____unaligned__™____builtin_alignof__c____sysapi__)____b
uiltin_isfloat__±____wchar_t__”___restrictÖÔ____int8×m____int16Øq____int32Ù▒____in
t64__ò____int128å#wchar_t_ë__inline(ù___cdecl.____based,,¥___stdcallÂP__declspec__Ö
___syscall__w___near__1__far_____far16______huge_____fortran_____export__.___pasca
l©]___tryª)__except«ì___finally·V___leave-
$__asm)‡__fastcallÖÜ___int8×ƒ__int16Ø‡__int32Ù•__int64*_€dllimport_Ø_€dllexport_³_
€naked_3_€thread_·_€allocate6__strset2__setjmpd__rotr^__rotl%o__outpwv__outpd___outp
~__lrotrx__lrotl___inpwõ__inpd”_inp/__exception_info-_exception_codeQ__enable___disa
bleª__alloca¨__abnormal_termination,___ull_rshiftØ<€]___ll_rshiftE___ll_lshiftb__emuluû_
_emul+__ReturnAddressð4€-_atan2Ž_atanž_asin»acos_abs_cosh˜?€bcos¤exp‘_fmod§_fabsx
_inpwe_inpd¼inpï_log1086€çlog,_labsÄ_memsetômemcpyßmemcmp%o_outpwv_outpdr_ou
tp7_pow<_strsetÆ_strlenl_strcpyW_strcmp+_strcat._sqrt’_sinhF_sinR_tanh_>€6_tanždefine
d____formalÇ_...Äabstract declaratorƒ_main‡_WinMain__DllMain»_.alignment
member.z___unnamed,=€$$C.drectve¤_.local static
guard.öCatch®___LOCAL_SIZE!___LINE__Var]___LINE__′____FILE__¥____DATE__
M___TIME__ù____TIMESTAMP__¨___INTEGRAL_MAX_BITSú___M_IX86
__MSC_VER,___WIN32{___MT¹__WIN32Í__NDEBUG8___WINDOWSÈ_J:\Vb60\Line2
1\Parity.cÄz€Ü{€_{€P{€({€ {€x{€ìz€_|€ð{€,|€È{€Øz€d{€<{€′{€Œ{€{€_|€p-
€Ø’€”Ž€Ô‘€¸Ž€_□€h‘€È“€à□€
□€˜’€t□€˜Ž€øŽ€„Ž€A@cc@ABB@CC@LL@DD@____€A_A@_c_`|€`|€xŒ€¼¢€H_Ex|€
ÿ_`”€¸”€_‘€ä”€|‘€4”€À•€Ü-€_-€°-€D“€4□€h˜€D™€Œ—
€ì’€ì˜€˜€_™€<˜€¬’€ˆ□€H□€_ˆ€`—€ô□€Œ□€`□€l’€′□€Ü“€Àˆ€_”€<•€„-€ä—
€Œ”€”˜€<‘€_□€¸—€4—€_—€_•€@’€_’€œ“€ä□€¸□€è‘€p“€D-
€”•€h•€i•€`|€__xŽ€c_ˆŽ€_€¬Ž€__>€□€__£___¬Ž€xŽ€A_üŽ€_>€_˜Ž€__Û□_<>€x□€___
c@_ˆŽ€_L>€x□€__Ÿ__\>€ä□€__°_B@_ˆŽ€_l>€$□€__á_D@_ˆŽ€_|>€˜Ž€__©__Œ>€$□€
__¢__œ>€$□€__¡__¬>€ä□€__¯__¼>€$□€__à__Ì>€˜Ž€__¨__Ü>€_l‘€__Ú€€_üŽ€_ô>€_$□
€__Ù€_?€Ø‘€__±€_ˆŽ€__?€Ø‘€__²_0?€_l‘€
_Ø□_@?€_˜Ž€!_×€_`?€œ’€"_ê_L@_ˆŽ€_x?€Ü’€#_ë_L_ˆŽ€_Œ?€œ’€$_é__?€œ’€%_í__°?
€Ü’€&_ì__À?€_l‘€'_â€_Ø?€Ì“€(_…____ˆŽ€_è?€Ì“€)_ƒ__ø?€Ì“€*_□___@€Ì“€+_□___@€˜
Ž€,____$@€Ì“€-
_%o___“€4@€Ì“€._o__@@€Ì“€/_a__L@€Ì“€0____\@€Ì“€1____l@€ä□€2_¯A_|@€$□€3_à
A_Œ@€˜Ž€4_¨A_˜@€Ì“€5_i__¨@€Ì“€6_e__′@€t-
€7___D_ˆŽ€_Ä@€l‘€8_-__Ô@€l‘€9_¬__ä@€˜Ž€:_ª__ô@€ä□€;_°B__A€x□€<_áB__A€˜Ž
€=_©B_$A€Ì“€>_®__0A€□€?_£B_@A€x□€@_§__PA€□€A_¤__`A€˜Ž€B_¥__pA€□€C_¦
__€A€Ì“€D_]__□A€Ì“€E_‡__ A€Ì“€F_k__¬A€Ì“€G_‹__ÌŽ€¼A€Ì“€H_s___Èz€____Èz€P
```

```
@¢€¨¢€ø¡€È¡€_¢€X¢€l¢€è¡€Ø¡€€¢€”¢€_¢€,¢€ðB€ý_C€ü_C€û,C€ú@C€øXC€64DtC€400D
„C€1000D˜C€1D¨C€1D´C€1DÄC€1DÔC€1D_èC€__ä¢€ü¢€_£€(£€<£€T£€h£€C:\MSDEV\
INCLUDEC:\MSDEV\MFC\includeK:\MSDEV\INCLUDEK:\MSDEV\MFC\includeD:\MS
DEV\INCLUDED:\MSDEV\MFC\include__Œ£€¬£€Ä£€Ð£€à£€è£€ì£€ô£€ü£€_INTEGRA
L_MAX_BITS=64_M_IX86=400_MSC_VER=1000_WIN32_MTWIN32NDEBUG_WIND
OWS
```

File: Parity.obj

```
L__ûl™9Œ__.drectve(d
_.data_Œ@@À-defaultlib:LIBCMT -defaultlib:OLDNAMES €__ƒ_…†__‰Š_Œ
_□_‘’“”__—˜__›_□ž_
¡¢#¤%&§¨)*«,-®/°12³4µ¶78¹º;¼=>¿@ÁÂCÄEFÇÈIJËLÍÎOÐQRÓTÕÖWXÙÚ[Ü]^ßàabãdå
æghéêkìmnïpñòsôuv÷øyzû|ýþ▒€__ƒ_…†__‰Š_Œ
_□_‘’“”__—˜__›_□ž_
¡¢#¤%&§¨)*«,-®/°12³4µ¶78¹º;¼=>¿@ÁÂCÄEFÇÈIJËLÍÎOÐQRÓTÕÖWXÙÚ[Ü]^ßàabãdå
æghéêkìmnïpñòsôuv÷øyzû|ýþ▒.fileþÿg_J:\Vb60\Line21\Parity.c.drectve___(.data_____parit
y___
```

File: stable.obj

```
L__
m™9d_-.drectve(d
_.datax_Œ__p@@À-defaultlib:LIBCMT -defaultlib:OLDNAMES ,__
_+__
__*__)_$__($_,)_4
_8&_@__D"_H!_P_T(_X__`
_h,_p_t+_|
_€*_„)_Œ__□$_”)_œ
_&_¨__¬"_°!_,_¼ _À__È
_Ì__Ð__Ø
_Ü__à__è
_ì__ð__ø__ü_______
__________
_____ ___(_
_,___0___4___<_
_@___D___L_
_P_(_X_
_`_,_h___l_(_p_'_x_ _€_,_ˆ_
_Œ_(_□_#_˜_ _ _&_¨___¬_*_°_)_,___¼_$_À_)_È_
_Ì_"_Ð_!_Ø__Ü_(_à___è_ _ð_(_ô_%_ü_
__,______+_______*___)_$___(_$_,_)_4_
_8_&_@___D_"_H_!_P__T___X___\___d___h_(_l___t___.fileþÿg_J:\Vb60\Line21\stable.c
.drectve___(.data___x_p_____ð____ ____-€___;`___Jh__X___XDSData _c _s __CRTest
__PACTest _, _” _ _® _, _Â _Î __URLTest _Ù _è _õ ___ ___ ___ _'_ _4_ __XDSCont _A_
__XDSEnd _N_ _Z_ _d_ _q_ _{_ _‰_ _”_
_ž__URLDataState_XDSEndState_XDSDataState_XDSContState_XDSStartState_ControlSt
ate_DataState_CtrlChangeTest_ChannelChange_EndOfCaptionTest_TextDecode_TextMode
Test_DumpText_DataSave_URLEndTest_URLDecode_PaintOnDecode_PaintOnTest_RollU
pDecode_RollUpTest_PopOnDecode_PopOnTest_ControlSave_ControlTest_XDSContTest_
XDSEndTest_XDSStart_DefaultTest_XDSBegin_XDSStartTest_F1XDSTest_NullTest
```

File: Tests.obj

```
L_#_m™9e
Z.drectve(Œ_
_.text_´_ _P`.debug$F_°_Ê__H__B.text
Ô_ _P`.debug$F_Þ_î__H__B.text_ø_ _P`.debug$F______H__B.text_)_
_P`.debug$F_B_R__H__B.text_\_ _P`.debug$F_u_…__H__B.text
□_ _P`.debug$F_œ_¬__H__B.text_¶_ _P`.debug$F_Ë_Û__H__B.text5å_
_P`.debug$F___*__H__B.text54_ _P`.debug$F_i_y__H__B.text0ƒ_
_P`.debug$F_³_Ã__H__B.text0Í_ _P`.debug$F_ý_
__H__B.text0__ _P`.debug$F_G_W__H__B.text&a_ _P`.debug$F_‡_—__H__B.text;¡_
_P`.debug$F_Ü_ì__H__B.text0ö_ _P`.debug$F_&  6        _H__B.text+@
_P`.debug$F_k         {         _H__B.textÆ…         _P`.debug$F_K
[
_H__B-defaultlib:LIBCMT -defaultlib:OLDNAMES ‚_Ã____€|$___À÷ØÃ
__€|$__s
ƒ|$__u_‚_Ã3ÀÃ____ŠD$_„Àt_<_s
¨_t_‚_Ã3ÀÃ____ŠD$_„Àt_<_s
¨_u_‚_Ã3ÀÃ____ŠD$_‚_<__À÷ØÃ
_ _ŠD$_<_v
<_s_‚_Ã3ÀÃ__%_‹D$_‹L$_‹D□üƒø_t_ƒø_t
ƒø_t_ƒø_uŠD$_<*t_<+t_3ÀÃ‚_Ã5_*_‹D$_‹L$_‹D□üƒø_t_ƒø_t
ƒø_t_ƒø_u_ŠD$_<$v
<(s_‚_Ã3ÀÃ5_/_‹L$_‹D$_‹D˜üƒø_t_ƒø_t
ƒø_t_ƒø_u
€|$_)u_‚_Ã3ÀÃ0_4_‹D$_‹L$_‹D□üƒø_t_ƒø_t
ƒø_t_ƒø_u
€|$_ u_‚_Ã3ÀÃ0_9_‹D$_‹L$_‹D□üƒø_t_ƒø_t
ƒø_t_ƒø_u
€|$_/u_‚_Ã3ÀÃ0_>_‹D$_‹L$_‹D□üƒø_~_ƒø }
€|$_?v_‚_Ã3ÀÃ&_C_‹L$_‹D$_ŠT$_□D˜ü€ú_‹H_‹v_€ú@s_ƒø_}_…Éu
ƒø_~_ƒù_t_‚_Ã3ÀÃ;_H_‹D$_‹L$_‹D□üƒø_t_ƒø_t
ƒø_t_ƒø_u
€|$_-
u_‚_Ã3ÀÃ0_M_‹L$_‹D$_ƒù_‹D˜üu_ƒø_u_ŠD$_<*t_<+u_‚_Ã3ÀÃ+_R_‹L$_‹D$_I‹ÑÁâ_Á
á
‹”_$_ƒú_□„____Ž›€|$_]_…□_Â€xû[_…„ŠHü€ù0|_€ù9~_€ùA|_€ùF~
€ùa|h€ùf▒cŠHý€ù0|_€ù9~_€ùA|_€ùF~
€ùa|G€ùf▒BŠHþ€ù0|_€ù9~_€ùA|_€ùF~
€ùa|&€ùf▒!Š@ÿ<0|_<9~_<A|_<F~_<a|
<f▒_‚_Ã3ÀÃÆ_W_.fileþÿg_J:\Vb60\Line21\Tests.c.drectve___(.text______
_.debug$F______.text___
___ _.debug$F______.text______ _.debug$F______.text____&_ _.debug$F
        _____.text
____4
```

```
_.debug$F_____
_.text__
_A _.debug$F
_____.text_____M_ _.debug$F_______.text___5_Z_ _.debug$F_______.text___5_h_
_.debug$F_______.text___0_t_ _.debug$F_______.text___0_□_
_.debug$F_______.text___0_Œ_ _.debug$F_______.text___&__PACTest_
_.debug$F_______.text___;_ž_ _.debug$F_______.text___0__CRTest_
_.debug$F_______.text __+__URLTest _.debug$F!_____.text"__Æ_®"
_.debug$F#____"_°_DefaultTest_NullTest_F1XDSTest_XDSStartTest_XDSContTest_XDS
EndTest_ControlTest_TextModeTest_RollUpTest_PaintOnTest_PopOnTest_EndOfCaptionT
est_CtrlChangeTest_URLEndTest
```

09054505 .050904

File: XDCMessages.obj

```
L___m™9ê    :.drectve(ä_
_.text______ _P`.data_9_@_0À.data_<_@_0À.debug$F_A_Q__H__B.textý_[_X_

_P`.data_Ú_@_0À.data_ì_@_0À.data_ï_@_0À.data_ò_@_0À.data_õ_@_0À.data_ø_@_0À
.data
ü_@_0À.data
__@_0À.debug$F___ __H__B.texth_*_'    _ _P`.data_Î    @_0À.debug$F_Ð    à
      _H__B-defaultlib:LIBCMT -defaultlib:OLDNAMES ‹
SV¸‹T$W‹\$_U‰
Š@_ˆB_‹ƒH_Áà__ƒL_□@+À□<M¹ÿÿÿÿò®÷Ñ+ù‹é‹÷¹ÿÿÿÿ‹ú+Àò®O‹ÍÁé_ó¥‹Íƒá_ó¤¿¹ÿÿÿ
ÿ+Àò®÷Ñ+ù‹é‹÷¹ÿÿÿÿ‹ú+Àò®O‹ÍÁé_ó¥‹Íƒá_ó¤‹ƒH_Áà__ƒL_Áà_□_@‹Œ_T__ËÆ„_X_‹ƒ
H_Áà__ƒL_Áà_□@+À□¼_X_¹ÿÿÿÿò®÷Ñ+ù‹Ù‹÷¹ÿÿÿÿ‹ú+Àò®O‹ËÁé_ó¥‹Ë]ƒá_ó¤_^[Ã_
_
_6__d
_ XDS
______ƒì_,S‹_VW‹T$_U‹l$_‰MŠ@_ˆE_‹,H_Áà__,L_□@+À□<M¹ÿÿÿÿò®÷Ñ+ù‹Ù‹÷¹ÿÿ
ÿÿ‹ý+Àò®O‹ËÁé_ó¥‹Ëƒá_ó¤‹,H_Áà__,L_Áà_□@Šœ_X_□__Š„_Y_ˆD$_ŠÃ$_Àè_„À_„ð_
<__„è_<__…¨_¿¹ÿÿÿÿ+Àò®÷Ñ+ù‹Ñ‹÷¹ÿÿÿÿ‹ý+Àò®O‹ÊÁé_ó¥‹Ê3Òƒá_ó¤ŠT$_¹ÿÿÿÿƒâ_+
À‹<•ò®÷Ñ+ù‰L$_‹÷¹ÿÿÿÿ‹ý+Àò®O‹L$_Áé_ó¥‹L$_€ã Àë_ƒá_ó¤ŠD$_$8
ØŠD$_Àû_$
ØÀû_"šöÃ_t0¿¹ÿÿÿÿ+Àò®÷Ñ+ù‹Ñ‹÷¹ÿÿÿÿ‹ý+Àò®O‹ÊÁé_ó¥‹Êƒá_ó¤öÃ_t0¿¹ÿÿÿÿ+Àò®÷
Ñ+ù‹Ñ‹÷¹ÿÿÿÿ‹ý+Àò®O‹ÊÁé_ó¥‹Êƒá_ó¤öÃ_t0¿¹ÿÿÿÿ+Àò®÷Ñ+ù‹Ñ‹÷¹ÿÿÿÿ‹ý+Àò®O‹ÊÁ
é_ó¥‹Êƒá_ó¤öÃ_t0¿¹ÿÿÿÿ+Àò®÷Ñ+ù‹Ñ‹÷¹ÿÿÿÿ‹ý+Àò®O‹ÊÁé_ó¥‹Êƒá_ó¤öÃ__„Ù¿¹ÿÿÿÿ+
Àò®÷Ñ+ù‹Ñ‹÷¹ÿÿÿÿ‹ý+Àò®O‹ÊÁé_ó¥‹Ê]ƒá_ó¤_^[ƒÄ_Ã¿¹ÿÿÿÿ+Àò®÷Ñ+ù‹Ñ‹÷¹ÿÿÿÿ‹ý+
Àò®O‹ÊÁé_ó¥‹Ê]ƒá_ó¤_^[ƒÄ_Ã¿¹ÿÿÿÿ+Àò®÷Ñ+ù‹Ñ‹÷¹ÿÿÿÿ‹ý+Àò®O‹ÊÁé_ó¥‹Êƒá_ó¤3
À¹ÿÿÿÿŠÃƒà_‹<…+Àò®÷Ñ+ù‹Ñ‹÷¹ÿÿÿÿ‹ý+Àò®O‹ÊÁé_ó¥‹Êƒá_ó¤]_^[ƒÄ_Ã_
_6__-(_ï._;_-_E_%_z_"_¯___ä_______U___□_+_Ë_,_  OTHER - non-USA,D,L,S,V,FV
TVPG -   MPAA -
ý_______‹_SV¸‹L$WU‰_Š@_‹T$_ˆA_‹,H_Áà__,L_□@+À□<M¹ÿÿÿÿò®÷Ñ+ù‹Ù‹÷¹ÿÿÿÿ
‹|$_+Àò®O‹ËÁé_ó¥‹Ëƒá_ó¤¿¹ÿÿÿÿ+Àò®÷Ñ+ù‹Ù‹÷¹ÿÿÿÿ‹|$_+Àò®O‹ËÁé_ó¥‹Ëƒá_ó¤3Û‹,
H_Áà__,L_Áà_□@9œ_T__Ž¨‹,H_Áà__,L_Áà_□_@_Ã_¾Œ_X_+À‹<□€ÿÿÿ¹ÿÿÿÿò®÷Ñ+
ù‹é‹÷¹ÿÿÿÿ‹|$_+Àò®O‹ÍÁé_ó¥‹Íƒá_ó¤¿¹ÿÿÿÿ+Àò®÷Ñ+ù‹é‹÷¹ÿÿÿÿ‹|$_+Àò®O‹ÍÁé_Çó¥‹Íƒ
á_ó¤‹,H_Áà__,L_Áà_□@9œ_T__□Xÿÿÿ]_^[Ã_
_
_6__f
_ß7___6_ h____3_.fileþÿg_J:\Vb60\Line21\XDSMessages.c.drectve___(.text________Def_
_.data________.data_________XDSMes_.debug$F_______.text___ý_
__Vcp_ _.data_____3__.data________.data ____u _.data
____‹
_.data_____¡__.data____·_.data

__
_Î
_.data___
```

```
_ï________'__.debug$F_______.text___h____Typ_
_.data_____4____types_.debug$F_______I_??_C@_02IHLC@?5?5?$AA@??_C@_04EPN
H@XDS?5?$AA@??_C@_0BC@NEMK@?5?5OTHER?5?9?5non?9USA?$AA@??_C@_
02DFDA@?0D?$AA@??_C@_02GBII@?0L?$AA@??_C@_02MJKF@?0S?$AA@??_C
@_02MJGO@?0V?$AA@??_C@_03LEKC@?0FV?$AA@??_C@_09GMHB@?5?5TVPG
?5?9?5?$AA@??_C@_09NPEP@?5?5MPAA?5?9?5?$AA@_mpaaRatings_flagmask_tvpg
Ratings??_C@_01FCOA@?5?$AA@
```

File: XDSROM.obj

```
L_v
m™9¶io_.drectve(„_
_.data_¬_@_0À.data_´_@_0À.data_¼_@_0À.data_Ä_@_0À.data_Ì_@_0À.data_Ô_@_0À.
data_Ü_@_0À.data_ä_@_0À.data_ê_@_0À.data_ò_@_0À.data_ù_@_0À.data
_@_0À.data_
_@_0À.data   __@_0À.data___@_0À.data_ _@_0À.data
        +_@_0À.data_4_@_0À.data_<_@_0À.data_C_@_0À.data_K_@_0À.data_S_@_0À
.data_Z_@_0À.data_a_@_0À.data_h_@_0À.data_p_@_0À.data_w_@_0À.data
~_@_0À.data_‹_@_0À.data“_@_0À.data   Ÿ_@_0À.data
        ¨_@_0À.data_±_@_0À.data_¸_@_0À.data
        ¿_@_0À.data_È_@_0À.data_Ð_@_0À.data_Ö_@_0À.data
        á_@_0À.data_ê_@_0À.data_ò_@_0À.data_ú_@_0À.data_ÿ_@_0À.data___@_0À.d
ata_
_@_0À.data    __@_0À.data
__@_0À.data_#_@_0À.data1_@_0À.data=_@_0À.data_I_@_0À.data_P_@_0À.data_U_@
_0À.data_\_@_0À.data_b_@_0À.dataj_@_0À.data_v_@_0À.data_}_@_0À.data_ˆ_@_0À.
data_□_@_0À.data
”_@_0À.dataž_@_0À.data_ª_@_0À.data
        ²_@_0À.data_»_@_0À.data_À_@_0À.data_È_@_0À.data_Ð_@_0À.data_Õ_@_0À
.data
        Ý_@_0À.data_æ_@_0À.data_î_@_0À.data_ù_@_0À.dataÿ_@_0À.data___@_0À.d
ata___@_0À.data
__@_0À.data $_@_0À.data_-
_@_0À.data_5_@_0À.data_@_@_0À.data_G_@_0À.data_N_@_0À.data
V_@_0À.data`_@_0À.data  i_@_0À.data_r_@_0À.data
        }_@_0À.data_†_@_0À.data_□_@_0À.data
˜_@_0À.data  ¢_@_0À.data_«_@_0À.data_¹_@_0À.data_À_@_0À.data_Æ_@_0À.data
Í_@_0À.data_×_@_0À.data_Ü_@_0À.data_â_@_0À.data
ð_@_0À.data_ú_@_0À.data___@_0À.data___@_0À.data___@_0À.data___@_0À.data___
@_0À.data_ _@_0À.data
%_@_0À.data_/_@_0À.data_1_@_0À.data_7_@_0À.data_9_@_0À.data_?_@_0À.data_B_
@_0À.data_D_@_0À.dataÜ+H_$Bõ_@@À-defaultlib:LIBCMT -defaultlib:OLDNAMES
Roll4 Roll3 Roll2 Paint Pop
WesternWeatherVideoVarietyTravelTennisTechnicalTalkSuspenseSpecialSoap
OperaShoppingServiceSeriesScienceRomanceReviewRepeatRepairReadingRacingPublicProf
essionalProductPrerecordedPremierePoliticsPoliceNatureNationalMysteryMusicMiniseriesM
ilitaryMeetingMedicalMathLocalLiveLegalLanguageInterviewInternationalInstructionInform
ationHorrorHomeHockeyHobbyHistoryHigh
SchoolHealthGovernmentGolfGardenGame/QuizFund
RaiserForeignFootballFoodFictionFashionFarmFantasyExerciseEroticaElementaryDramaDoc
umentaryDanceCrimeContemporaryConsumerConcertCommentaryComedyCombatCollegeC
lassicalBusinessBulletinBasketballBaseballAwardsAutomobileAnthologyAnimatedAdvertise
mentActionOTHERSportsReligiousNewsMovieEntertainmentEducationTV-MA TV-14 TV-
```

PG TV-GTV-Y7 TV-YNoneNot RatedXNC-17RPG-13PGGN/A__ ___ _

Null StartDuratShow Type
VChipAudioCapti0108 Aspec010A 010B Comp1Comp2010E 010F
Desc1Desc2Desc3Desc4Desc5Desc6Desc7Desc80118 0119 011A 011B 011C 011D 011E
011F 0120 0121 0122 0123 0124 0125 0126 0127 0128 0129 012A 012B 012C 012D 012E
012F 0130 0131 0132 0133 0134 0135 0136 0137 0138 0139 013A 013B 013C 013D 013E
013F 0140 0141 0142 0143 0144 0145 0146 0147 0148 0149 014A 014B 014C 014D 014E
014F 0150 0151 0152 0153 0154 0155 0156 0157 0158 0159 015A 015B 015C 015D 015E
015F 0160 0161 0162 0163 0164 0165 0166 0167 0168 0169 016A 016B 016C 016D 016E
016F 0170 0171 0172 0173 0174 0175 0176 0177 0178 0179 017A 017B 017C 017D 017E
017F 0300 0301 0302 0303 0304 0305 0306 0307 0308 0309 030A 030B 030C 030D 030E
030F 0310 0311 0312 0313 0314 0315 0316 0317 0318 0319 031A 031B 031C 031D 031E
031F 0320 0321 0322 0323 0324 0325 0326 0327 0328 0329 032A 032B 032C 032D 032E
032F 0330 0331 0332 0333 0334 0335 0336 0337 0338 0339 033A 033B 033C 033D 033E
033F 0340 0341 0342 0343 0344 0345 0346 0347 0348 0349 034A 034B 034C 034D 034E
034F 0350 0351 0352 0353 0354 0355 0356 0357 0358 0359 035A 035B 035C 035D 035E
035F 0360 0361 0362 0363 0364 0365 0366 0367 0368 0369 036A 036B 036C 036D 036E
036F 0370 0371 0372 0373 0374 0375 0376 0377 0378 0379 037A 037B 037C 037D 037E
037F 0500 NetwkAffilDelay0504 0505 0506 0507 0508 0509 050A 050B 050C 050D 050E
050F 0510 0511 0512 0513 0514 0515 0516 0517 0518 0519 051A 051B 051C 051D 051E
051F 0520 0521 0522 0523 0524 0525 0526 0527 0528 0529 052A 052B 052C 052D 052E
052F 0530 0531 0532 0533 0534 0535 0536 0537 0538 0539 053A 053B 053C 053D 053E
053F 0540 0541 0542 0543 0544 0545 0546 0547 0548 0549 054A 054B 054C 054D 054E
054F 0550 0551 0552 0553 0554 0555 0556 0557 0558 0559 055A 055B 055C 055D 055E
055F 0560 0561 0562 0563 0564 0565 0566 0567 0568 0569 056A 056B 056C 056D 056E
056F 0570 0571 0572 0573 0574 0575 0576 0577 0578 0579 057A 057B 057C 057D 057E
057F 0700 Time ImpulSupplZone 0705 0706 0707 0708 0709 070A 070B 070C 070D 070E
070F 0710 0711 0712 0713 0714 0715 0716 0717 0718 0719 071A 071B 071C 071D 071E
071F 0720 0721 0722 0723 0724 0725 0726 0727 0728 0729 072A 072B 072C 072D 072E
072F 0730 0731 0732 0733 0734 0735 0736 0737 0738 0739 073A 073B 073C 073D 073E
073F OOBNB0741 0742 0743 0744 0745 0746 0747 0748 0749 074A 074B 074C 074D
074E 074F 0750 0751 0752 0753 0754 0755 0756 0757 0758 0759 075A 075B 075C 075D
075E 075F 0760 0761 0762 0763 0764 0765 0766 0767 0768 0769 076A 076B 076C 076D

```
076E 076F 0770 0771 0772 0773 0774 0775 0776 0777 0778 0779 077A 077B 077C 077D
077E 077F 0900 0901 0902 0903 0904 0905 0906 0907 0908 0909 090A 090B 090C 090D
090E 090F 0910 0911 0912 0913 0914 0915 0916 0917 0918 0919 091A 091B 091C 091D
091E 091F 0920 0921 0922 0923 0924 0925 0926 0927 0928 0929 092A 092B 092C 092D
092E 092F 0930 0931 0932 0933 0934 0935 0936 0937 0938 0939 093A 093B 093C 093D
093E 093F 0940 0941 0942 0943 0944 0945 0946 0947 0948 0949 094A 094B 094C 094D
094E 094F 0950 0951 0952 0953 0954 0955 0956 0957 0958 0959 095A 095B 095C 095D
095E 095F 0960 0961 0962 0963 0964 0965 0966 0967 0968 0969 096A 096B 096C 096D
096E 096F 0970 0971 0972 0973 0974 0975 0976 0977 0978 0979 097A 097B 097C 097D
097E 097F 0B00 WSCodWSMes0B03 0B04 0B05 0B06 0B07 0B08 0B09 0B0A 0B0B
0B0C 0B0D 0B0E 0B0F 0B10 0B11 0B12 0B13 0B14 0B15 0B16 0B17 0B18 0B19 0B1A
0B1B 0B1C 0B1D 0B1E 0B1F 0B20 0B21 0B22 0B23 0B24 0B25 0B26 0B27 0B28 0B29
0B2A 0B2B 0B2C 0B2D 0B2E 0B2F 0B30 0B31 0B32 0B33 0B34 0B35 0B36 0B37 0B38
0B39 0B3A 0B3B 0B3C 0B3D 0B3E 0B3F 0B40 0B41 0B42 0B43 0B44 0B45 0B46 0B47
0B48 0B49 0B4A 0B4B 0B4C 0B4D 0B4E 0B4F 0B50 0B51 0B52 0B53 0B54 0B55 0B56
0B57 0B58 0B59 0B5A 0B5B 0B5C 0B5D 0B5E 0B5F 0B60 0B61 0B62 0B63 0B64 0B65
0B66 0B67 0B68 0B69 0B6A 0B6B 0B6C 0B6D 0B6E 0B6F 0B70 0B71 0B72 0B73 0B74
0B75 0B76 0B77 0B78 0B79 0B7A 0B7B 0B7C 0B7D 0B7E 0B7F 0D00 0D01 0D02 0D03
0D04 0D05 0D06 0D07 0D08 0D09 0D0A 0D0B 0D0C 0D0D 0D0E 0D0F 0D10 0D11
0D12 0D13 0D14 0D15 0D16 0D17 0D18 0D19 0D1A 0D1B 0D1C 0D1D 0D1E 0D1F
0D20 0D21 0D22 0D23 0D24 0D25 0D26 0D27 0D28 0D29 0D2A 0D2B 0D2C 0D2D
0D2E 0D2F 0D30 0D31 0D32 0D33 0D34 0D35 0D36 0D37 0D38 0D39 0D3A 0D3B 0D3C
0D3D 0D3E 0D3F 0D40 0D41 0D42 0D43 0D44 0D45 0D46 0D47 0D48 0D49 0D4A 0D4B
0D4C 0D4D 0D4E 0D4F 0D50 0D51 0D52 0D53 0D54 0D55 0D56 0D57 0D58 0D59 0D5A
0D5B 0D5C 0D5D 0D5E 0D5F 0D60 0D61 0D62 0D63 0D64 0D65 0D66 0D67 0D68 0D69
0D6A 0D6B 0D6C 0D6D 0D6E 0D6F 0D70 0D71 0D72 0D73 0D74 0D75 0D76 0D77
0D78 0D79 0D7A 0D7B 0D7C 0D7D 0D7E 0D7F
_____k____k____k____k____j____i____k____k__
_k__$_k__(_k__,_k__0_k__4_k__8_k__<_k__@_k__D_k__H_k__L_k__P_k__T_k__X_k_
_\_k__`_k__d_k__h_k__l_k__p_k__t_k__x_k__|_k__€_k__„_k__ˆ_k__Œ_k__□_k__”_k__˜
_k__œ_k____k__¤_k__¨_k__¬_k__°_k__´_k__¸_k__¼_k__À_k__Ä_k__È_k__Ì_k__Ð_k__
Ô_k__Ø_k__Ü_k__à_k__ä_k__è_k__ì_k__ð_k__ô_k__ø_k__ü_k____k____k____k____k___
_k____k____k____k__
_k__$_k__(_k__,_k__0_k__4_k__8_k__<_k__@_k__D_k__H_k__L_k__P_k__T_k__X_k_
_\_k__`_k__d_k__h_k__l_k__p_k__t_k__x_k__|_k__€_k__„_k__ˆ_k__Œ_k__□_k__”_k__˜
_k__œ_k____k__¤_k__¨_k__¬_k__°_k__´_k__¸_k__¼_k__À_k__Ä_k__È_k__Ì_k__Ð_k__
Ô_k__Ø_k__Ü_k__à_k__ä_k__è_k__ì_k__ð_k__ô_k__ø_k__ü_k____k____k____k____k___
_k____k____k____k__
_k__$_k__(_k__,_k__0_k__4_k__8_k__<_k__@_k__D_k__H_k__L_k__P_k__T_k__X_k_
_\_k__`_k__d_k__h_k__l_k__p_k__t_k__x_k__|_k__€_k__„_k__ˆ_k__Œ_k__□_k__”_k__˜
_k__œ_k____k__¤_k__¨_k__¬_k__°_k__´_k__¸_k__¼_k__À_k__Ä_k__È_k__Ì_k__Ð_k__
Ô_k__Ø_k__Ü_k__à_k__ä_k__è_k__ì_k__ð_k__ô_k__ø_k__ü_k____k____k____k____k___
_k____k____k____k__
_k__$_k__(_k__,_k__0_k__4_k__8_k__<_k__@_k__D_k__H_k__L_k__P_k__T_k__X_k_
_\_k__`_k__d_k__h_k__l_k__p_k__t_k__x_k__|_k__€_k__„_k__ˆ_k__Œ_k__□_k__”_k__˜
_k__œ_k____k__¤_k__¨_k__¬_k__°_k__´_k__¸_k__¼_k__À_k__Ä_k__È_k__Ì_k__Ð_k__
```

```
Ô_k__Ø_k__Ü_k__à_k__ä_k__è_k__ì_k__ð_k__ô_k__ø_k__ü_k__ k___k___k__k___
k___k___k___k__ k__$k__(k__,k__0k__4k__8k__<k__@k__Dk__Hk__Lk__P
k__Tk__Xk__\k__`k__dk__hk__lk__pk__tk__xk__|k__€k__„k__ˆk__Œk__□k__”
k__˜k__œk__ k__¤k__¨k__¬k__°k__´k__¸k__¼k__Àk__Äk__Èk__Ìk__Ðk__Ô
k__Øk__Ük__àk__äk__èk__ìk__ðk__ôk__øk__ü
k__!k___!k___!k__!k___!k___!k___!k___!k__
!k__$!k__(!k__,!k__0!k__4!k__8!k__<!k__@!k__D!k__H!k__L!k__P!k__T!k__X!k__\!k__
_`!k__d!k__h!k__l!k__p!k__t!k__x!k__|!k__€!k__„!k__ˆ!k__Œ!k__□!k__”!k__˜!k__œ!k__
!k__¤!k__¨!k__¬!k__°!k__´!k__¸!k__¼!k__À!k__Ä!k__È!k__Ì!k__Ð!k__Ô!k__Ø!k__Ü!k__
_à!k__ä!k__è!k__ì!k__ð!k__ô!k__ø!k__ü!k__"k___"k___"k__"k___"k___"k___"k___"k__
"k__$"k__("k__,"k__0"k__4"k__8"k__<"k__@"k__D"k__H"k__L"k__P"k__T"k__X"k__\"
k__`"k__d"k__h"k__l"k__p"k__t"k__x"k__|"k__€"k__„"k__ˆ"k__Œ"k__□"k__”"k__˜"k__
œ"k__ "k__¤"k__¨"k__¬"k__°"k__´"k__¸"k__¼"k__À"k__Ä"k__È"k__Ì"k__Ð"k__Ô"k__Ø
"k__Ü"k__à"k__ä"k__è"k__ì"k__ð"k__ô"k__ø"k__ü"k__#k___#k___#k__#k___#k___#k__
_#k___#k__
#k__$#k__(#k__,#k__0#k__4#k__8#k__<#k__@#k__D#k__H#k__L#k__P#k__T#k__X#k__
_\#k__`#k__d#k__h#k__l#k__p#k__t#k__x#k__|#k__€#k__„#k__ˆ#k__Œ#k__□#k__”#k__˜
#k__œ#k__ #k__¤#k__¨#k__¬#k__°#k__´#k__¸#k__¼#k__À#k__Ä#k__È#k__Ì#k__Ð#k__
Ô#k__Ø#k__Ü#k__à#k__ä#k__è#k__ì#k__ð#k__ô#k__ø#k__ü#k__$k___$k___$k__$k___
$k___$k___$k___$k__
$k__$$k__($k__,$k__0$k__4$k__8$k__<$k__@$k__D$k__H$k__L$k__P$k__T$k__X$k__
_\$k__`$k__d$k__h$k__l$k__p$k__t$k__x$k__|$k__€$k__„$k__ˆ$k__Œ$k__□$k__”$k__˜
$k__œ$k__ $k__¤$k__¨$k__¬$k__°$k__´$k__¸$k__¼$k__À$k__Ä$k__È$k__Ì$k__Ð$k__
Ô$k__Ø$k__Ü$k__à$k__ä$k__è$k__ì$k__ð$k__ô$k__ø$k__ü$k__%k___%k___%k__%k__
__%k___%k___%k___%k__
%k__$%k__(%k__,%k__0%k__4%k__8%k__<%k__@%k__D%k__H%k__L%k__P%k__
T%k__X%k__\%k__`%k__d%k__h%k__l%k__p%k__t%k__x%k__|%k__€%k__„%k__ˆ%
k__Œ%k__□%k__”%k__˜%k__œ%k__ %k__¤%k__¨%k__¬%k__°%k__´%k__¸%k__¼%k
__À%k__Ä%k__È%k__Ì%k__Ð%k__Ô%k__Ø%k__Ü%k__à%k__ä%k__è%k__ì%k__ð%
k__ô%k__ø%k__ü%k__&k___&k___&k__&k___&k___&k___&k___&k__
&k__$&k__(&k__,&k__0&k__4&k__8&k__<&k__@&k__D&k__H&k__L&k__P&k__T
&k__X&k__\&k__`&k__d&k__h&k__l&k__p&k__t&k__x&k__|&k__€&k__„&k__ˆ&k__
Œ&k__□&k__”&k__˜&k__œ&k__ &k__¤&k__¨&k__¬&k__°&k__´&k__¸&k__¼&k__À
&k__Ä&k__È&k__Ì&k__Ð&k__Ô&k__Ø&k__Ü&k__à&k__ä&k__è&k__ì&k__ð&k__ô&
k__ø&k__ü&k__'k___'k___'k__'k___'k___'k___'k___'k__
'k__$'k__('k__,'k__0'k__4'k__8'k__<'k__@'k__D'k__H'k__L'k__P'k__T'k__X'k__\'k__`'k__
d'k__h'k__l'k__p'k__t'k__x'k__|'k__€'k__„'k__ˆ'k__Œ'k__□'k__”'k__˜'k__œ'k__ 'k__¤'k__¨'k
__¬'k__°'k__´'k__¸'k__¼'k__À'k__Ä'k__È'k__Ì'k__Ð'k__Ô'k__Ø'k__Ü'k__à'k__ä'k__è'k__ì'
k__ð'k__ô'k__ø'k__ü'k__(k___(k___(k__(k___(k___(k___(k___(k__
(k__$(k__((k__,(k__0(k__4(k__8(k__<(k__@(k__D(k__H(k__L(k__P(k__T(k__X(k__\(k__
_`(k__d(k__h(k__l(k__p(k__t(k__x(k__|(k__€(k__„(k__ˆ(k__Œ(k__□(k__”(k__˜(k__œ(k__
(k__¤(k__¨(k__¬(k__°(k__´(k__¸(k__¼(k__À(k__Ä(k__È(k__Ì(k__Ð(k__Ô(k__Ø(k__Ü(k__
_à(k__ä(k__è(k__ì(k__ð(k__ô(k__ø(k__ü(k__)k___)k___)k__)k___)k___)k___)k___)k__
)k__$)k__()k__,)k__0)k__4)k__8)k__<)k__@)k__D)k__H)k__L)k__P)k__T)k__X)k__\)k__
_`)k__d)k__h)k__l)k__p)k__t)k__x)k__|)k__€)k__„)k__ˆ)k__Œ)k__□)k__”)k__˜)k__œ)k__
)k__¤)k__¨)k__¬)k__°)k__´)k__¸)k__¼)k__À)k__Ä)k__È)k__Ì)k__Ð)k__Ô)k__Ø)k__Ü)k__
_à)k__ä)k__è)k__ì)k__ð)k__ô)k__ø)k__ü)k__*`___*]___*Z__*W___*T___*Q___*N___*
K__*H__$*E__(*B__,*?__0*<__4*9__8*6__<*H__H*3__L*0__P*-
__T**__X*'__\*$__`*!__d*___h*___l*___p*___t*___x*___|*__€*
      __„*___ˆ*___Œ*__□*ý_”*ú_˜*÷_œ*ô_ *ñ_¤*î_¨*ë_¬*è_°*å_´*â_¸*ß_¼*Ü_À*Ù_
Ä*Ö_È*Ó_Ì*Ð_Ð*Í_Ô*Ê_Ø*Ç_Ü*Ä_à*Á_ä*¾_è*»_ì*¸_ð*µ_ô*²_ø*¯_ü*¬_+©__+¦__+£_
+ __+□__+š__+—__+”_
+‘_$+Ž_(+‹_,+ˆ_0+…_4+‚_8+▒_<+|_@+y_D+v_H+s_L+p_P+m_T+j_X+g_\+d_`+a_d+^_h
+[_l+X_p+U_t+R_x+O_|+L_€+I_„+F_ˆ+C_Œ+@_□+=_”+:_˜+7_œ+4_ +1_¤+._¨++_¬+(_°
+%_´+"_¸+__¼+__À+__Ä+__È+__Ì+__Ð+
_Ô+
_Ø+__.fileþÿg_J:\Vb60\Line21\XDSROM.c.drectve___(.data________.data_____
__.data_____<__.data_____W__.data_____s__.data_____‘__.data_____«__.data ____Å
        _.data
____Ý
_.data_____÷__.data_______.data

__
_)_
_.data_____E___.data___   _\___.data_____w___.data_____‘___.data___
      ¯___.data_____Ê___.data_____ä___.data_____ý___.data_________.data_____1__
_.data_____J___.data_____c___.data_____|___.data_____–___.data_____¯___.data___
_È___.data_____è___.data________.data __      _ _ _.data!__
      _;_!_.data"____V_"_.data#____o_#_.data$__
      _ˆ_$_.data%____£_%_.data&____½_&_.data'____Õ_'_.data(__
      _ó_(_.data)______)_.data*____(_*_.data+____B_+_.data,____Y_,_.data-____q_-
_.data.____ˆ_._.data/__      __/_.data0__
_»_0_.data1____×_1_.data2___ø_2_.data3_____3_.data4____6_4_.data5____O_5_.data6__
__f_6_.data7____~_7_.data8____–
_8_.data9___°_9_.data:____Ð_:_.data;____é_;_.data<______<_.data=______=_.data>__
_7_>_.data?___T_?_.data@____t_@_.dataA__
      _Ž_A_.dataB____©_B_.dataC____À_C_.dataD____Ú_D_.dataE____ô_E_.dataF___
___F_.dataG__
      _%_G_.dataH____@_H_.dataI____Z_I_.dataJ____x_J_.dataK___□_K_.dataL____¯
_L_.dataM____Ç_M_.dataN__
_ß_N_.dataO__
      _ÿ_O_.dataP______P_.dataQ____3_Q_.dataR____Q_R_.dataS____j_S_.dataT____ƒ
_T_.dataU__
_□_U_.dataV__      _¹_V_.dataW__      _Ô_W_.dataX____ï_X_.dataY__    _
      Y_.dataZ____(      Z_.data[____A      [_.data\__
__    \_.data]__    _{    ]_.data^____–^_.data_____·__.data`____Ð`_.dataa____è
      a_.datab__

_
b_.datac_____
c_.datad____3
d_.datae____K
e_.dataf__
_l
```

```
f_.datag____ˆ
g_.datah____£
h_.datai____¾
i_.dataj____Ù
j_.datak____ï
k_.datal____
_l_.datam____!_m_.datan__
_7_n_.datao____T_o_.datap____h_p_.dataq____□_q_.datar____•_r_.datas____®_s_.datat_
___Ã_t_.datau____×_u_.datav__Ü+õ__CTypeÈ+v__typesH*v_î_@*v_ø_
*v__*v__XDSDisp_v__Vcp __Typ __Def
__XDSMes_v__€_v__v_(??_C@_07OJHM@Roll4?5?5?$AA@??_C@_07FHBN@Roll3?5
?5?$AA@??_C@_07FAL@Roll2?5?5?$AA@??_C@_07ILCI@Paint?5?5?$AA@??_C@_0
7LFNA@Pop?5?5?5?5?$AA@??_C@_07PCEP@Western?$AA@??_C@_07GGMB@Weat
her?$AA@??_C@_05HAHE@Video?$AA@??_C@_07GOLO@Variety?$AA@??_C@_06
MAKO@Travel?$AA@??_C@_06IECD@Tennis?$AA@??_C@_09MAPH@Technical?$A
A@??_C@_04MNKN@Talk?$AA@??_C@_08EABE@Suspense?$AA@??_C@_07GALJ
@Special?$AA@??_C@_0L@NHB@Soap?5Opera?$AA@??_C@_08HFCK@Shopping?$
AA@??_C@_07KNHF@Service?$AA@??_C@_06NHCF@Series?$AA@??_C@_07LHB
D@Science?$AA@??_C@_07HHNK@Romance?$AA@??_C@_06HNJB@Review?$AA
@??_C@_06NPIB@Repeat?$AA@??_C@_06NNPN@Repair?$AA@??_C@_07NCCF@R
eading?$AA@??_C@_06DPOE@Racing?$AA@??_C@_06GCHD@Public?$AA@??_C@
_0N@EONA@Professional?$AA@??_C@_07KKIB@Product?$AA@??_C@_0M@BOC
@Prerecorded?$AA@??_C@_08JKHO@Premiere?$AA@??_C@_08BGDL@Politics?$AA
@??_C@_06KJAF@Police?$AA@??_C@_06BIAE@Nature?$AA@??_C@_08CKMM@N
ational?$AA@??_C@_07GPIF@Mystery?$AA@??_C@_05PHIM@Music?$AA@??_C@_
0L@IEGN@Miniseries?$AA@??_C@_08LHGP@Military?$AA@??_C@_07FDIO@Meeti
ng?$AA@??_C@_07HHJI@Medical?$AA@??_C@_04LHAK@Math?$AA@??_C@_05P
HNP@Local?$AA@??_C@_04HLML@Live?$AA@??_C@_05NPLB@Legal?$AA@??_C
@_08DFDG@Language?$AA@??_C@_09IAGO@Interview?$AA@??_C@_0O@FJHI@I
nternational?$AA@??_C@_0M@PJEI@Instruction?$AA@??_C@_0M@LKAF@Informati
on?$AA@??_C@_06KKNE@Horror?$AA@??_C@_04OGPG@Home?$AA@??_C@_06K
GO@Hockey?$AA@??_C@_05LHBI@Hobby?$AA@??_C@_07JFNM@History?$AA@?
?_C@_0M@PJHA@High?5School?$AA@??_C@_06MACP@Health?$AA@??_C@_0L@
JHGG@Government?$AA@??_C@_04NPML@Golf?$AA@??_C@_06CMMI@Garden?$
AA@??_C@_09CNCP@Game?1Quiz?$AA@??_C@_0M@ENFA@Fund?5Raiser?$AA@?
?_C@_07JNNB@Foreign?$AA@??_C@_08NPEO@Football?$AA@??_C@_04JEEI@Foo
d?$AA@??_C@_07IHEG@Fiction?$AA@??_C@_07MCOL@Fashion?$AA@??_C@_04I
FFI@Farm?$AA@??_C@_07OCKK@Fantasy?$AA@??_C@_08JEHB@Exercise?$AA@?
?_C@_07NCAH@Erotica?$AA@??_C@_0L@JBPC@Elementary?$AA@??_C@_05CGP
O@Drama?$AA@??_C@_0M@NOLC@Documentary?$AA@??_C@_05FKAK@Dance?$
AA@??_C@_05HMON@Crime?$AA@??_C@_0N@KIIC@Contemporary?$AA@??_C@
_08JELE@Consumer?$AA@??_C@_07DLE@Concert?$AA@??_C@_0L@LGND@Com
mentary?$AA@??_C@_06IGJL@Comedy?$AA@??_C@_06CAIJ@Combat?$AA@??_C
@_07LGJO@College?$AA@??_C@_09JFNA@Classical?$AA@??_C@_08FMGD@Busin
ess?$AA@??_C@_08FKAB@Bulletin?$AA@??_C@_0L@PKKG@Basketball?$AA@??_
C@_08GPNH@Baseball?$AA@??_C@_06IPOB@Awards?$AA@??_C@_0L@CFFG@A
```

```
utomobile?$AA@??_C@_09IADM@Anthology?$AA@??_C@_08HEBO@Animated?$AA
@??_C@_0O@PINI@Advertisement?$AA@??_C@_06NMFG@Action?$AA@??_C@_05
FGEM@OTHER?$AA@??_C@_06BIM@Sports?$AA@??_C@_09IEGM@Religious?$A
A@??_C@_04KJCI@News?$AA@??_C@_05CHKH@Movie?$AA@??_C@_0O@JFFK
@Entertainment?$AA@??_C@_09HOLD@Education?$AA@??_C@_06HOCB@TV?9MA
?5?$AA@??_C@_06BJIF@TV?914?5?$AA@??_C@_06KCEN@TV?9PG?5?$AA@??_C
@_04HI@TV?9G?$AA@??_C@_06HFLA@TV?9Y7?5?$AA@??_C@_04CMC@TV?9Y?
$AA@??_C@_04COF@None?$AA@??_C@_09HKPO@Not?5Rated?$AA@??_C@_01FI
AI@X?$AA@??_C@_05DMNK@NC?917?$AA@??_C@_01FJJO@R?$AA@??_C@_05
MAGO@PG?913?$AA@??_C@_02DACJ@PG?$AA@??_C@_01PACF@G?$AA@??_C
@_03PHGM@N?1A?$AA@_flagmask_tvpgRatings_mpaaRatings_MinLength_MaxLength
```

Appendix C

Copyright © 2000 I LOVE TV INC. (868970 Ontario Inc.)
All rights reserved unless otherwise noted

Cold Fusion

Showtime

File: AddComment.cfm

```
<cfif ParameterExists(Edit)>

        <cfif Len(Trim(NewComment)) GT 0>

                <!--- Get the max of EventID's --->
                <cfquery name="GetMax"
                datasource="VBIContent"
                username="caption"
                password="trewq55">
                select Max(Eventid)as maxid from ctlfile
                </cfquery>

                <cfset newEvent = '#incrementValue(GetMax.maxid)#'>

                <!--- Insert a new record and pass EventID --->

                <cfquery name="AddEvent"
                datasource="VBIContent"
                username="caption"
                password="trewq55">
                Insert into ctlfile
                (Eventid,
                Seconds,
                Type,
                Clipsource,
                 Comment

                )
                Values
                (#newEvent#,
                #LTime#,
                'C',
                '#Clipsrc#',
                 '#NewComment#'
                )
                </cfquery>

        </cfif>
<!--- Close Window --->
<!--- <body onLoad="window.close();"> --->
<cfoutput>
<cflocation url="./thankyou.cfm?LTime=#Ltime#&DS=VBIContent&Clipsrc=#ClipSrc#">

</cfoutput>
</cfif>


<font face="Arial" size="-1">Type in your comment, then press "Save"<br>
<!--- Use LTime for current time --->
<form action="AddComment.cfm?Edit=Yes" method="POST"
onSubmit="setTimeout('window.close()',2000);">
<cfoutput>
<input type="Hidden" name="LTime" value="#LTime#">
<cfif ParameterExists(EventID)>
<input type="Hidden" name="EventID" value="#EventID#">
</cfif>
<input type="Hidden" name="DS" value="#DS#">
<input type="Hidden" name="Clipsrc" value="#Clipsrc#">
</cfoutput>
<center>
<textarea name="NewComment" cols="40" rows="2" wrap="VIRTUAL"></textarea>
<input type="Submit" value="Save"></center>
</font>
</form>
```

09851605.050901

File: banner2.cfm

```
<html>
<head>
<title></title>
</head>
<SCRIPT LANGUAGE = "JavaScript">
<!-- hide this script from old browsers
LinkWindow="status=no,toolbar=yes,directories=no,menubar=yes,location=yes,scrollbars=y
es,resizable=yes,HEIGHT=400,WIDTH=600";
// -->
</script>
<body bgcolor="#5B94D1">
        <CFIF ParameterExists(imgsrc) AND ParameterExists(msg2)>
        <cfoutput>
        <A href="http://#msg2#" target="NetWin"
onClick="window.open(",'NetWin',LinkWindow);"><IMG align="left" alt="#caption#"
border=0 src="http://#imgsrc#"></a>
        </cfoutput>
        <CFELSE>
        &nbsp;
        </CFIF>

</body>
</html>
```

File: buttonbar2.cfm

```
<html>
<head>
<title></title>
</head>

<SCRIPT LANGUAGE = "JavaScript">
name = 'mymainwindow';
CaptionWin=null;
<!-- hide this script from old browsers
LinkWindow="status=no,toolbar=yes,directories=no,menubar=yes,location=no,scrollbars=yes,resizable=yes,HEIGHT=480,WIDTH=640";
SelectWindow="status=no,toolbar=yes,directories=no,menubar=yes,location=no,scrollbars=yes,resizable=yes,HEIGHT=400,WIDTH=600";
function WinOpen()
{
        if (CaptionWin == null)
        {
                CaptionWin = window.open("caption2.cfm", "CaptionWin",
"menubar=0,toolbar=0,resizeable=0,width=320,height=360");
                CaptionWin.focus();
                document.forms[0].isWinOpen.value="true";
                return false;
        }
        else
        {
                if (CaptionWin.closed == true)
                {
                        CaptionWin = window.open("caption2.cfm", "CaptionWin",
"menubar=0,toolbar=0,resizeable=0,width=320,height=360");
                        CaptionWin.focus();
                        document.forms[0].isWinOpen.value="true";
                        return false;
                }
        }
}
//function isWinOpen()
//{
//      if (CaptionWin == null)
//      {
//              WinOpen();
//              CaptionWin.hide();
//              return false;
//      }
```

09951505.050901

```
//        else
//        {
//                  if (CaptionWin.closed == true)
//                  {
//                            return false;
//                  }
//                  else
//                  {
//                                      return true;
//                  }
//        }
//}
// -->
</script>

<!--- buttonbar in frame name="ButtonWin" --->

<body bgcolor="#8FBC8F" text="Black" link="Blue" vlink="Blue" alink="#FF4500">
<!--- <cfif Channel is ">
onLoad="window.open('../Selector/index.cfm','SelectWin',LinkWindow);"
</cfif> --->
<table width='100%' border="0" cellspacing="0" cellpadding="0" bgcolor="#8FBC8F">
                  <tr>
                  <td width="10" rowspan="3"
background="../images/site/transparent.gif"></td>
                  <td height="3" colspan="2" background="../images/site/transparent.gif"></td>
                  <td rowspan="3" align="RIGHT" valign="TOP"><img
src="/IMAGES/site/buttonbar_curve.jpg" border=0 alt="" align="top" width=14
height=15></td>
                  </tr>
                  <tr>
                  <td><font face="Arial" size="-1">
                                    <a href="../Selector/channels.cfm" target="SelectWin"
onClick="window.open(",'SelectWin',LinkWindow);">Channels</a>
                           </td><td>
                           <font face="Arial" size="-1">
                                    <a
href="../Links/index.cfm?ClipSrc=<cfoutput>#Clipsrc#&DS=#DS#</cfoutput>"
target="OutLinkWin" onClick="window.open(",'OutLinkWin',LinkWindow);">Links</a>
                           </td>

                  </tr>
                  <tr><td>
                                    <font face="Arial" size="-1">
```

```
                                        <font face="Arial" size="-1"> <a
href="../Selector/selections.cfm" target="SelectWin"
onClick="window.open('','SelectWin',LinkWindow);">Selections</a>
                </td><td>
                                <font face="Arial" size="-1">
                                <cfif NOT Channel is ''>
                                        <a
href="../chat/index.cfm?Channel=<cfoutput>#Channel#</cfoutput>" target="OutLinkWin"
onClick="window.open('','OutLinkWin',LinkWindow);">Chat</a>
                                </cfif>
                        </td>
                </tr>
<!---           <tr>
                <td colspan="2">
                                <font face="Arial" size="-1">
                                <cfif NOT Channel is ''>
                                        <a
href="caption2.cfm?Channel=<cfoutput>#Channel#</cfoutput>" target="CaptionWin"
onClick="return(WinOpen())">CC</a>
                                </cfif>
                </td>
                </tr> --->
<tr><td height="3" colspan="4" background="../images/site/transparent.gif">
</td></tr>

<tr><td colspan="4">
        <table width="100%"  border="0" cellspacing="0" cellpadding="2">
        <form name='SearchForm' action="../Links/search.cfm" target="OutLinkWin"
onSubmit="window.open('','OutLinkWin',LinkWindow);
setTimeout('document.SearchForm.reset()',2000);" method="POST">
        <input type="Hidden" name="isWinOpen" value="">
        <tr bgcolor="#8FBC8F"><td align="CENTER">
                <font face="Arial" size="1">
                Find <input type="Text" name="SrchFor" size="15" maxlength="50"></font>
                <center><font size="1"><input type="Submit"
value="Search!"></font></center>
        </td></tr>
                </form>
        </table>

</td></tr>
</table>
</body>
</html>
```

09851605.050901

File: caption2.cfm

```
<html>
<head>
<title>Closed Captioning</title>
<script language="JavaScript">
function SetUnLoad()
{
        opener.document.forms[0].isWinOpen.value="false";
        window.close();
}
</script>
</head>
<body bgcolor="ghostwhite">
<!--- sponsor in frame name="SponsorWin" --->
<table width="100%" border="0" cellspacing="0" cellpadding="0" bgcolor="ghostwhite">
<tr>
<form name="cc">
<td align='center'>
        <table width="95%" border="0" cellspacing="0" cellpadding="0"><tr>
        <td>
        <CFIF ParameterExists(caption)>
        <cfoutput>
        <font size="-1">
        <textarea name='Caption' cols='40' rows='10'
wrap="VIRTUAL">#Replace(caption,"||","","ALL")#</textarea>
        </font>
        </cfoutput>
        </cfif>
        </td>
        </tr>
        </table>
</td>
</TR>
</form>
</TABLE>
</body>
</html>
```

09851605.050901

File: comment2.cfm

```
<html>
<head>
<title></title>
</head>
<SCRIPT LANGUAGE = "JavaScript">
<!-- hide this script from old browsers
CommWin=null;
LinkWindow="status=no,toolbar=yes,directories=no,menubar=yes,location=yes,scrollbars=yes,resizable=yes,HEIGHT=400,WIDTH=600";
SmallWindow="status=no,toolbar=yes,directories=no,menubar=no,location=no,scrollbars=no,resizable=yes,HEIGHT=200,WIDTH=400";

function WinOpen()
{
	var ltime=parent.DataWin.document.Comments.ltime.value;
	var ltime_1=Math.floor(ltime/1000);
	var clip=parent.DataWin.document.Comments.Clipsrc.value;
	var ds=parent.DataWin.document.Comments.DS.value;
	var What_url="addcomment.cfm?LTime="+ltime_1+"&ds="+ds+"&Clipsrc="+clip;
	if (CommWin == null)
	{
		CommWin = window.open(What_url, "CommWin", SmallWindow);
		CommWin.focus();
	}
	else
	{
		if (CommWin.closed == true)
		{
			CommWin = window.open(What_url, "CommWin", SmallWindow);
			CommWin.focus();
		}
	}
}


function SetElapseTime()
{
	var ltime=parent.DataWin.document.Comments.ltime.value;
	var clip=parent.DataWin.document.Comments.Clipsrc.value;
	var ds=parent.DataWin.document.Comments.DS.value;
	location.href="addcomment.cfm?LTime="+ltime+"&ds="+ds+"&Clipsrc="+clip;
}
// -->
```

```
</script>
<body bgcolor="#F5F5F5" link="Blue" text="Black" vlink="#5B94D1" alink="#FF4500">
<font face="Arial" size="-2">
&nbsp;&nbsp;Comments: &nbsp; &nbsp; &nbsp; &nbsp;
<!---
<CFIF ParameterExists(LTime)>
<a href="addcomment.cfm?LTime=#LTime#" target="CommWin"
onClick="window.open('','CommWin',SmallWindow);SetElapseTime();">
<CFELSE>
<a href="addcomment.cfm?LTime=0" target="CommWin"
onClick="window.open('','CommWin',SmallWindow);SetElapseTime();">
</CFIF>
--->
<a href="#" onClick="WinOpen()">
Add Your Comment</a><br>
</font>
<font face="Arial" size="1">
<CFIF ParameterExists(msg)>
<cfoutput>
#Replace(msg,"||","","ALL")#
</cfoutput>
</cfif>
</font>
</body>
</html>
```

09851605.050901

File: community.cfm

```
<body bgcolor="ivory">

community in frame name="CommunityWin"
```

09851505.050601

File: data2.cfm

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
	<title>Untitled</title>
</head>
<body>
<form name="Sponsor" action="sponsor2.cfm" method="post" enctype="application/x-
www-form-urlencoded" target="SponsorWin">
<input type="Hidden" name="imgsrc" value="">
<input type="Hidden" name="msg2" value="">
<input type="Hidden" name="caption" value="">
<input type="Hidden" name="Net" value="">
<input type="Hidden" name="Channel" value="">
</FORM>

<form name="Links" action="linklist2.cfm" method="post" enctype="application/x-www-
form-urlencoded" target="LinkListWin">
<input type="Hidden" name="msg" value="">
</FORM>

<form name="Ad" action="banner2.cfm" method="post" enctype="application/x-www-form-
urlencoded" target="BannerWin">
<input type="Hidden" name="imgsrc" value="">
<input type="Hidden" name="msg2" value="">
<input type="Hidden" name="caption" value="">
</FORM>

<form name="Caption" action="caption2.cfm" method="post" enctype="application/x-www-
form-urlencoded" target="CaptionWin">
<input type="Hidden" name="caption" value="">
</FORM>

<form name="Comments" action="comment2.cfm" method="post" enctype="application/x-
www-form-urlencoded" target="CommentWin">
<input type="Hidden" name="ltime" value="0">
<input type="Hidden" name="msg" value="">
<CFOUTPUT>
<input type="Hidden" name="Clipsrc" value="#Clipsrc#">
<input type="Hidden" name="DS" value="#ds#">
</CFOUTPUT>
</FORM>
</body>
```

09851505.050901

</html>

09951605.050901

File: linklist2.cfm

```
<html>
<head>
<title></title>
<SCRIPT LANGUAGE = "JavaScript">
<!-- hide this script from old browsers
LinkWindow="status=no,toolbar=yes,directories=no,menubar=yes,location=yes,scrollbars=y
es,resizable=yes,HEIGHT=400,WIDTH=600";
// -->
</script>
</head>
<body bgcolor="#F4A460" link="Blue" text="Black" vlink="#5B94D1" alink="#FF4500">
<font face="Arial" size="1">
&nbsp;&nbsp;Current Links:
<br>
<b>
<CFIF ParameterExists(msg)>
<cfoutput>
#Replace(msg,"||","","ALL")#
</cfoutput>
</cfif>
</font>
</body>
</html>
```

09951605.050901

File: main2.cfm

```
<cfparam name="Channel" default="">
<cfparam name="Net" default="">
<cfparam name="Clipsrc" default="">
<cfparam name="size" default="">
<cfparam name="DS" default="Adserver">

<script language="JavaScript">
<!--
<!--- if (document.images) --->
   self.name = 'showtime';
//-->
 </script>
<!-- frames -->
<frameset rows="*,0,62" framespacing="0" border="no" frameborder="0" marginwidth="0"
marginheight="0" bordercolor="white">
  <frameset  cols="169,*">
                <frameset  rows="132,90,*">
                <frame name="SponsorWin"
src="sponsor2.cfm?Channel=<cfoutput>#Channel#&Net=#Net#&ClipSrc=#Clipsrc#&DS=#
DS#</cfoutput>" framespacing="0" border="0" frameborder="0" marginwidth="0"
marginheight="0" bordercolor="white" scrolling="No" noresize>
                        <frame name="ButtonWin"
src="buttonbar2.cfm?Channel=<cfoutput>#Channel#&Net=#Net#&Clipsrc=#Clipsrc#&Size
=#Size#&DS=#DS#</cfoutput>" framespacing="0" border="0" frameborder="0"
marginwidth="0" marginheight="0" bordercolor="white" scrolling="no" frameborder="no">
                        <frame name="LinkListWin"
src="linklist2.cfm?<cfoutput>Clipsrc=#Clipsrc#&DS=#DS#</cfoutput>" framespacing="0"
border="0" frameborder="0" marginwidth="0" marginheight="0" bordercolor="white"
scrolling="no" noresize frameborder="no">
                </frameset>
                 <frame name="StreamWin"
src="stream2.cfm?Channel=<cfoutput>#Channel#&Net=#Net#&Clipsrc=#Clipsrc#&Size=#
Size#&DS=#DS#</cfoutput>" framespacing="0" border="0" frameborder="0"
marginwidth="0" marginheight="0" bordercolor="white" scrolling="no" frameborder="no">
        </frameset>
   <frame name="DataWin"
src="data2.cfm?Channel=<cfoutput>#Channel#&Net=#Net#&Clipsrc=#Clipsrc#&Size=#Siz
e#&DS=#DS#</cfoutput>" framespacing="0" border="0" frameborder="0" marginwidth="0"
marginheight="0" bordercolor="white" scrolling="no" frameborder="no">
        <frameset  cols="*,468">
                <frame name="CommentWin"
src="comment2.cfm?<cfoutput>Clipsrc=#Clipsrc#&DS=#DS#</cfoutput>"
```

```
framespacing="0" border="0" frameborder="0" marginwidth="0" marginheight="0"
bordercolor="white" scrolling="no" noresize frameborder="no">
                <frame name="BannerWin"
src="banner2.cfm?<cfoutput>Clipsrc=#Clipsrc#&DS=#DS#</cfoutput>" framespacing="0"
border="0" frameborder="0" marginwidth="0" marginheight="0" bordercolor="white"
scrolling="no" noresize frameborder="no">
        </frameset>

</frameset>
```

File: main3.cfm

```
<cfparam name="Channel" default="">
<cfparam name="Net" default="">
<cfparam name="Clipsrc" default="">
<cfparam name="size" default="">
<cfparam name="DS" default="Adserver">

<script language="JavaScript">
<!--
<!--- if (document.images) --->
   self.name = 'showtime';
//-->
 </script>
<!-- frames -->
<frameset rows="*,62" framespacing="0" border="0" frameborder="0" marginwidth="0"
marginheight="0" bordercolor="white">
  <frameset  cols="169,*,0">
                 <frameset  rows="132,130,*">
                 <frame name="SponsorWin"
src="sponsor2.cfm?Channel=<cfoutput>#Channel#&Net=#Net#&ClipSrc=#Clipsrc#&DS=#
DS#</cfoutput>" framespacing="0" border="0" frameborder="0" marginwidth="0"
marginheight="0" bordercolor="white" scrolling="No" noresize>
                       <frame name="ButtonWin"
src="buttonbar2.cfm?Channel=<cfoutput>#Channel#&Net=#Net#&Clipsrc=#Clipsrc#&Size
=#Size#&DS=#DS#</cfoutput>" framespacing="0" border="0" frameborder="0"
marginwidth="0" marginheight="0" bordercolor="white" scrolling="no" frameborder="no">
                       <frame name="LinkListWin"
src="linklist2.cfm?<cfoutput>Clipsrc=#Clipsrc#&DS=#DS#</cfoutput>" framespacing="0"
border="0" frameborder="0" marginwidth="0" marginheight="0" bordercolor="white"
scrolling="no" noresize frameborder="no">
                </frameset>
   <frame name="StreamWin"
src="stream3.cfm?Channel=<cfoutput>#Channel#&Net=#Net#&Clipsrc=#Clipsrc#&Size=#
Size#&DS=#DS#</cfoutput>" framespacing="0" border="0" frameborder="0"
marginwidth="0" marginheight="0" bordercolor="white" scrolling="no" frameborder="no">
   <frame name="DataWin"
src="data2.cfm?Channel=<cfoutput>#Channel#&Net=#Net#&Clipsrc=#Clipsrc#&Size=#Siz
e#&DS=#DS#</cfoutput>" framespacing="0" border="0" frameborder="0" marginwidth="0"
marginheight="0" bordercolor="white" scrolling="no" frameborder="no">
   </frameset>
       <frameset  cols="*,468">
       <frame name="CommentWin"
src="comment2.cfm?<cfoutput>Clipsrc=#Clipsrc#&DS=#DS#</cfoutput>"
```

```
framespacing="0" border="0" frameborder="0" marginwidth="0" marginheight="0"
bordercolor="white" scrolling="no" noresize frameborder="no">
        <frame name="BannerWin"
src="banner2.cfm?<cfoutput>Clipsrc=#Clipsrc#&DS=#DS#</cfoutput>" framespacing="0"
border="0" frameborder="0" marginwidth="0" marginheight="0" bordercolor="white"
scrolling="no" noresize frameborder="no">

        </frameset>
</frameset>
```

File: main4.cfm

```
<cfparam name="Channel" default="">
<cfparam name="Net" default="">
<cfparam name="Clipsrc" default="">
<cfparam name="size" default="">
<cfparam name="DS" default="Adserver">

<script language="JavaScript">
<!--
<!--- if (document.images) --->
   self.name = 'showtime';
//-->
 </script>
<!-- frames -->
<frameset rows="*,62" framespacing="0" border="0" frameborder="0" marginwidth="0"
marginheight="0" bordercolor="white">
  <frameset  cols="169,*,0">
                  <frameset  rows="132,130,*">
                  <frame name="SponsorWin"
src="sponsor2.cfm?Channel=<cfoutput>#Channel#&Net=#Net#&ClipSrc=#Clipsrc#&DS=#
DS#</cfoutput>" framespacing="0" border="0" frameborder="0" marginwidth="0"
marginheight="0" bordercolor="white" scrolling="No" noresize>
                           <frame name="ButtonWin"
src="buttonbar2.cfm?Channel=<cfoutput>#Channel#&Net=#Net#&Clipsrc=#Clipsrc#&Size
=#Size#&DS=#DS#</cfoutput>" framespacing="0" border="0" frameborder="0"
marginwidth="0" marginheight="0" bordercolor="white" scrolling="no" frameborder="no">
                           <frame name="LinkListWin"
src="linklist2.cfm?<cfoutput>Clipsrc=#Clipsrc#&DS=#DS#</cfoutput>" framespacing="0"
border="0" frameborder="0" marginwidth="0" marginheight="0" bordercolor="white"
scrolling="no" noresize frameborder="no">
                   </frameset>
   <frame name="StreamWin"
src="stream4.cfm?Channel=<cfoutput>#Channel#&Net=#Net#&Clipsrc=#Clipsrc#&Size=#
Size#&DS=#DS#</cfoutput>" framespacing="0" border="0" frameborder="0"
marginwidth="0" marginheight="0" bordercolor="white" scrolling="no" frameborder="no">
   <frame name="DataWin"
src="data2.cfm?Channel=<cfoutput>#Channel#&Net=#Net#&Clipsrc=#Clipsrc#&Size=#Siz
e#&DS=#DS#</cfoutput>" framespacing="0" border="0" frameborder="0" marginwidth="0"
marginheight="0" bordercolor="white" scrolling="no" frameborder="no">
   </frameset>
        <frameset  cols="*,468">
        <frame name="CommentWin"
src="comment2.cfm?<cfoutput>Clipsrc=#Clipsrc#&DS=#DS#</cfoutput>"
```

```
framespacing="0" border="0" frameborder="0" marginwidth="0" marginheight="0"
bordercolor="white" scrolling="no" noresize frameborder="no">
          <frame name="BannerWin"
src="banner2.cfm?<cfoutput>Clipsrc=#Clipsrc#&DS=#DS#</cfoutput>" framespacing="0"
border="0" frameborder="0" marginwidth="0" marginheight="0" bordercolor="white"
scrolling="no" noresize frameborder="no">

          </frameset>
</frameset>
```

09851605.050901

File: search.cfm

```
<html>
<head>
<title>Search</title>
</head>
<SCRIPT LANGUAGE="JavaScript">
<!--
 window.name="OutLinkWin";
 //-->
 </SCRIPT>

<body onLoad="if (window.focus) self.focus(); else alert('Page Loaded');">



search Window


<form>
<input type="Button" value="Close" onClick="window.close();">
</form>

</body>
</html>
```

09851605.050901

File: selector.cfm

09851505.050901

File: sponsor2.cfm

```
<cfquery name="GetNet" datasource="TV">
Select* from NetStns
Where Call = '#Net#'
</cfquery>

<cfquery name="GetChannel" datasource="TV">
Select* from NetStns
Where Call = '#Channel#'
</cfquery>

<html>
<head>
<title>sponsor</title>

</head>
<SCRIPT LANGUAGE = "JavaScript">
<!-- hide this script from old browsers
LinkWindow="status=no,toolbar=yes,directories=no,menubar=yes,location=yes,scrollbars=y
es,resizable=yes,HEIGHT=400,WIDTH=600";
// -->
function setSponsor(msg1)
{
document.writeln(msg1);
}
</script>

<body bgcolor="#228B22">
<!--- sponsor in frame name="SponsorWin" --->
<table width="169" border="0" cellspacing="0" cellpadding="0" bgcolor="#228B22">
<tr>
<td width="30" rowspan="4" bgcolor="#8FBC8F">
<IMG align="middle" alt="" border=0 height=132 src="/images/site/sponsor_curve.gif"
width=30></a>
</TD>

<td height="3" bgcolor="#228b22" background="transparent.gif">
</td>

</TR>
<tr>
<td>
        <table width="120" border="0" cellspacing="0" cellpadding="0">
        <tr>
```

09851605.050901

```
          <td width="32" height="32">
          <cfif GetNet.recordcount gt 0>
          <cfoutput>
          <a href="http://#GetNet.LinkURL#" target="NetWin"
onClick="window.open(",'NetWin',LinkWindow);"><IMG align="middle"
alt="#GetNet.LinkText#" border=0 height=32 src="#getNet.ImageLoc#" width=32></a>
          </cfoutput>
          <cfelse>
          &nbsp;
          </cfif>
          </td>

          <td>
          <a href="http://ilovetv.ca" target="new"><IMG align="middle" alt="" border=0
height=32 src="/images/ads/network/love.jpg" width=32></a>
          </td>
          <td width="32" height="32">
          <cfif GetChannel.recordcount GT 0>
          <cfoutput>
          <a href="http://#GetChannel.LinkURL#" target="NetWin"
onClick="window.open(",'NetWin',LinkWindow);"><IMG align="middle"
alt="#GetChannel.LinkText#" border=0 height=32 src="#getChannel.ImageLoc#"
width=32></a>
          </cfoutput>
          <cfelse>
          &nbsp;
          </cfif>
          </td>
          </tr>
          </table>
</td>
</TR>
<tr>
<td width="120" height="90">
          <CFIF ParameterExists(imgsrc) AND ParameterExists(msg2)>
          <cfoutput>
          <A href="http://#msg2#" target="NetWin"
onClick="window.open(",'NetWin',LinkWindow);"><IMG align="left" alt="#caption#"
border=0 src="http://#imgsrc#"></a>
          </cfoutput>
          <CFELSE>
          &nbsp;
          </CFIF>
</TD>
</tr>
<tr>
```

```
<td height="3" bgcolor="#228b22" background="transparent.gif">

</td>
</TR>
</TABLE>
</body>
</html>>
```

09851605.050901

File: stream2.cfm

09851605.050901

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
        <title>Untitled</title>
</head>

<cfif Trim(Channel) is "or channel is ' '>
<body bgcolor="#228B22" leftmargin=3 topmargin=3>
<!---
        <a href="http://home.sol.no/~ggunners/NetBird.html">Net Bird</a>
         --->
        <br><br><br>
        <table width="80%" align="CENTER" valign="MIDDLE" bgcolor="#F4A460">
        <tr><td>
        <font face="Arial" size="+1" color="Blue">Select a program to view.<br><br>
        Selections button presents archived shows for this demo.<br><br>
        Channels button presents a channel selector mock-up.<br><br>
        Links button shows all the links from the current program.<br><br>
        Almost everything on the screen is a live link to the internet.
</td>
</tr>
        </table>
         &nbsp; <br>


        </center>
        </font>

</body>
</html>
<cfelse>



<script language="javascript">
<!--
var count = 0;
var doPolling;
var Ver4 = parseInt(navigator.appVersion) >= 4;
var Nav4 = ((navigator.appName == "Netscape") && Ver4);
var IE4  = ((navigator.userAgent.indexOf("MSIE") != -1) && Ver4);
var Messages = ";
var msg=";
var Current_time=0;
//caption
var c_msg=";
var c_count=0;
//link
var l_count=0;
var l_msg=";
//comment
var m_count=0;
var m_msg=";

var isAppletReady=false;
CaptionWin=null;
<CFOUTPUT>
var Net='#Net#';
var Channel='#Channel#';
</CFOUTPUT>
// Save window content dimensions to work around resize
// behaviour in WIN32 versions of Navigator 4.
if (Nav4)
{
        var origWidth = window.innerWidth;
        var origheight = window.innerHeight;
}

function WinOpen()
{
        if (CaptionWin == null)
        {
                CaptionWin = window.open("caption2.cfm", "CaptionWin",
"menubar=0,toolbar=0,resizeable=0,width=320,height=180");
                CaptionWin.focus();
                return false;
        }
        else
        {
                if (CaptionWin.closed == true)
                {
                        CaptionWin = window.open("caption2.cfm", "CaptionWin",
"menubar=0,toolbar=0,resizeable=0,width=320,height=360");
                        CaptionWin.focus();
                        return false;
                }
        }
}
// Invoked by <body> tag's onResize event handler to refresh the page for
// Navigator 4 to assist in getting positioned elements back in place.
function RestorePage()
{
        if (Nav4)
        {
                if (origWidth != window.innerWidth || origHeight != window.innerHeight)
                        history.go(0);
        }
}


function SendChatMessage(Author, Type, Message)
{
        document.applets[0].SendMessage(Author, Type, Message);
}

function SendElapsedTimeToApplet(Time)
{
        //var tx=Time/1000;
        //var tx1=Math.floor(tx);
        document.applets[0].SendElapsedTime(Time);
}


function HandleSessionMessageEx(Author, Type, Message)
{
        if (IE4)
        {
        document.forms[0].WhatPosition.value=Message;
        }
        else if (Nav4)
        {
        document.forms[0].WhatPosition.value=Message;

        }

}

function HandleAppletLoad(Message)
{
                isAppletReady=true;
}

function HandleElapsedTime(Message)
```

```
{
        if (IE4)
        {
        document.forms[0].WhatPosition.value=Message;
        }
        else if (Nav4)
        {
        document.forms[0].WhatPosition.value=Message;

        }

}
function HandleSponsor(Message1,Message2,Message3)
{
        //Handle the images in the SponsorWin frame
        parent.DataWin.document.Sponsor.imgsrc.value = Message1;
        parent.DataWin.document.Sponsor.msg2.value = Message2;
        parent.DataWin.document.Sponsor.caption.value = Message3;
        parent.DataWin.document.Sponsor.Net.value = Net;
        parent.DataWin.document.Sponsor.Channel.value = Channel;

        HandleLinks(Message2,Message3,"");

        parent.DataWin.document.Sponsor.submit();
}
function HandleLinks(Message1,Message2, Message3)
{
        var Type=Message3;
        if(Type == "old")
        {
                l_msg="<a href='http://"+Message1+"' target='NetWin'
onClick=\"window.open('','NetWin',LinkWindow)\">"+Message2+"</a><br>||"+l_msg;
        }
        else
        {
                l_msg="<a href='http://"+Message1+"' target='NetWin'
onClick=\"window.open('','NetWin',LinkWindow)\">"+Message2+"</a><br>||"+l_msg;
        }

        l_count++;

        if(l_count >5)
        {
                        var last=l_msg.lastIndexOf("||");
                        l_msg = l_msg.substring(0,last);
        }
```

```
		parent.DataWin.document.Links.msg.value = l_msg;
		parent.DataWin.document.Links.submit();



	}
	function HandleAd(Message1,Message2, Message3)
	{
		//Handle the images in the SponsorWin frame
		parent.DataWin.document.Ad.imgsrc.value = Message1;
		parent.DataWin.document.Ad.msg2.value = Message2;
		parent.DataWin.document.Ad.caption.value = Message3;
		HandleLinks(Message2,Message3,"");
		parent.DataWin.document.Ad.submit();
	}
	function HandleCaption(msg)
	{
		if(msg !='')
		{
			if(c_count >5)
			{
				var last=c_msg.lastIndexOf("||");
				c_msg = c_msg.substring(0,last);
				c_msg=msg+"\r\n||"+c_msg;
			}
			else
			{
				c_msg=msg+"\r\n||"+c_msg;
			}
			//Handle the images in the SponsorWin frame
			if(CaptionWin == null || CaptionWin.closed == true)
			{

			}
			else
			{
				parent.DataWin.document.Caption.caption.value = c_msg;
				parent.DataWin.document.Caption.submit();
			}
		c_count++;
		}
	}


	function HandleComments(msg)
	{
```

```
          if(msg !='')
          {
                    if(m_count >5)
                    {
                              var last=m_msg.lastIndexOf("||");
                              m_msg = m_msg.substring(0,last);
                              m_msg=msg+"<BR>||"+m_msg;
                    }
                    else
                    {
                              m_msg=msg+"<BR>||"+m_msg;
                    }
          parent.DataWin.document.Comments.msg.value = m_msg;
          parent.DataWin.document.Comments.submit();
          m_count++;
          }

}


function HandleSessionErrorEx(ErrNumber, ErrDescription)
{
          alert(ErrDescription);
}


function SendElapsedTime()
{
          if(document.video1.GetPlayState()== 3)
          {
            // Make sure that the applet has been loaded
            if(isAppletReady)
            {
                              var ElpTime=document.video1.GetPosition();
                              if(ElpTime != '')
                              {
                              SendElapsedTimeToApplet(ElpTime);
                              parent.DataWin.document.Comments.ltime.value = ElpTime;
                              }
                              polling();
                    }
                    else
                    {
                              alert("The Applet on your browser has not been started. Please refresh
your page");
                              stopPolling();
```

```
                }
            }
            else
            {
        doPolling = setTimeout("SendElapsedTime()",500);
            }
    }

    function stopPolling()
    {
        clearTimeout(doPolling)
    }
    function polling()
    {

        doPolling = setTimeout("SendElapsedTime()",1000);
    }

    //-->
    </script>

    <body bgcolor="#228B22" leftmargin=3 topmargin=3>

    <applet id="ChatApplet" width="1" height="1" code="ChatClient.class"
    codebase="http://blanche.tcn.net:8090/" VIEWASTEXT mayscript>
    <param name="MessageHandler" value="HandleSessionMessageEx">
    <param name="ElapsedTimeMessageHandler" value="HandleElapsedTime">
    <param name="ErrorHandler" value="HandleSessionErrorEx">
    <param name="SponsorHandler" value="HandleSponsor">
    <param name="LinksHandler" value="HandleLinks">
    <param name="CaptionHandler" value="HandleCaption">
    <param name="AdHandler" value="HandleAd">
    <param name="CommentsHandler" value="HandleComments">
    <param name="CaptionHandler" value="HandleCaption">
    <param name="ErrorHandler" value="HandleSessionErrorEx">
    <param name="TimeHandler" value="HandleTime">
    <param name="isAppletLoad" value="HandleAppletLoad">

    <CFOUTPUT>
    <param name="ClipSource" value="#ClipSrc#">
    </CFOUTPUT>
    </applet>
    <script Language="JavaScript">
    polling();
    </script>
```

```
<table width="100%" border="0" cellspacing="0" cellpadding="0" align="LEFT"
valign="TOP">
<tr>
<td align="left" valign="top">
        <!--- stream in frame name="streamWin" --->

        <cfparam name="Channel" default="">
        <cfparam name="Net" default="">
        <cfparam name="Clipsrc" default="">
        <cfparam name="size" default="">

        <cfif Size is 'S'>
                <cfset Width=320>
                <cfset Height=240>
        <cfelseif Size is 'L'>
                <cfset Width=512>
                <cfset Height=384>
                <cfelseif Size is 'X'>
                <cfset Width=240>
                <cfset Height=180>
        </cfif>

        <cfif Not Clipsrc is ">
                <cfif ClipSrc is "SunPar3" and Size is "L">
                        <cfset cliploc =
'http://www.broadcast.com/business/sun/98par3challenge_102099_100.ram'>
                <cfelseif ClipSrc is "SunPar3" and Size is "S">
                        <cfset cliploc =
'http://www.broadcast.com/business/sun/98par3challenge_102099_56.ram'>

                <cfelseif ClipSrc is "Live001" and Size is "S">
                        <cfset cliploc = '/media/56k/Live001.ram'>

                <cfelseif ClipSrc is "Auction" and Size is "S">
                        <cfset cliploc = '/media/56k/auction.ram'>
                <cfelse>
                        <cfif Size is 'S'>
                                <cfset cliploc = '/media/56k/#Clipsrc#.ram'>
                        <cfelseif Size is 'L'>
                                <cfset cliploc = '/media/108k/#Clipsrc#.ram'>
                        <cfelse>
                                <cfset cliploc = '/media/#Clipsrc#.ram'>
                        </cfif>
                </cfif>
        </cfif>
```

09851605.050901

```
          <center>
          <cfoutput>
          <object
           ID=video1
           CLASSID="clsid:CFCDAA03-8BE4-11cf-B84B-0020AFBBCCFA"
           height=#Height#
           width=#Width#
           >

           <param name="controls" value="ImageWindow">
           <param NAME="console" VALUE="Clip1">
           <param NAME="autostart" VALUE="true">
           <param NAME="src" VALUE="#Cliploc#">
           <embed name="video1" SRC="#Cliploc#"
           type="audio/x-pn-realaudio-plugin"
           CONSOLE="Clip1"
           controls="ImageWindow"
           height=#Height#
           width=#Width#
           AUTOSTART=true>
           </object>

           <BR>
           <form>
           <input type="Button" name="Play_Button" value="Play"
onClick="document.applets[0].init();document.video1.DoPlay();polling()">
           <input type="Button" name="Pause_Button" value="Pause"
onClick="document.video1.DoPause();stopPolling();">
           <input type="Button" name="Stop_Button" value="Stop"
onClick="document.video1.DoStop();stopPolling();">
<!---      <input type="Button" name="position" value="Position"
onClick="javascript:alert(document.video1.GetPosition())">
           <input type="Button" name="buffer" value="Send"
onClick="SendChatMessage('ripu', 'ChatMessage', document.video1.GetPosition())">
           <input type="Text" name="WhatPosition"> --->
          </form>
           </cfoutput>
          </center>
</td>
</tr>
<tr>
<td colspan="2">
                              <font face="Arial" size="-1">
                                   <a
href="caption2.cfm?Channel=<cfoutput>#Channel#</cfoutput>" target="CaptionWin"
```

```
onClick="return(WinOpen())"><img src="/images/site/CC.gif" width=28 height=23
border=1 alt="Closed Captioning" align="right"></a>
</td>
</tr>
</table>
</body>
</html>
</cfif>
```

File: stream3.cfm

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
        <title>Untitled</title>
</head>

<cfif Trim(Channel) is "or channel is ' '>
<body bgcolor="#228B22" leftmargin=3 topmargin=3>

        <a href="http://home.sol.no/~ggunners/NetBird.html">Net Bird</a>

        <br><br><br>
        <table width="80%" align="CENTER" valign="MIDDLE" bgcolor="#F4A460">
        <tr><td>
        <font face="Arial" size="+1" color="Blue">Select a program to view.<br><br>
        Selections button presents archived shows for this demo.<br><br>
        Channels button presents a channel selector mock-up.<br><br>
        Links button shows all the links from the current program.<br><br>
        Almost everything on the screen is a live link to the internet.
</td>
</tr>
        </table>
         &nbsp; <br>


        </center>
        </font>

</body>
</html>
<cfelse>


<script language="javascript">
<!--
var count = 0;
var doPolling;
var Ver4 = parseInt(navigator.appVersion) >= 4;
var Nav4 = ((navigator.appName == "Netscape") && Ver4);
var IE4  = ((navigator.userAgent.indexOf("MSIE") != -1) && Ver4);

var Messages = ";
var msg=";
```

09851605.050901

```
var c_msg=";
var Current_time=-1000;
CaptionWin=null;
var c_count=0;
// Save window content dimensions to work around resize
// behaviour in WIN32 versions of Navigator 4.
if (Nav4)
{
        var origWidth = window.innerWidth;
        var origheight = window.innerHeight;
}


// Invoked by <body> tag's onResize event handler to refresh the page for
// Navigator 4 to assist in getting positioned elements back in place.
function RestorePage()
{
        if (Nav4)
        {
                if (origWidth != window.innerWidth || origHeight != window.innerHeight)
                        history.go(0);
        }
}


function SendChatMessage(Author, Type, Message)
{
        document.applets[0].SendMessage(Author, Type, Message);
}

function SendElapsedTimeToApplet(Time)
{
        document.applets[0].SendElapsedTime(Time);
}


function HandleSessionMessageEx(Author, Type, Message)
{
        if (IE4)
        {
        document.forms[0].WhatPosition.value=Message;
        }
        else if (Nav4)
        {
        document.forms[0].WhatPosition.value=Message;
```

```
          }

     }
     function HandleElapsedTime(Message)
     {
          if (IE4)
          {
          document.forms[0].WhatPosition.value=Message;
          }
          else if (Nav4)
          {
          document.forms[0].WhatPosition.value=Message;

          }

     }
     function HandleSponsor(Message1,Message2,Message3)
     {
          //Handle the images in the SponsorWin frame
          parent.DataWin.document.Sponsor.imgsrc.value = Message1;
          parent.DataWin.document.Sponsor.msg2.value = Message2;
          parent.DataWin.document.Sponsor.caption.value = Message2;
          parent.DataWin.document.Sponsor.submit();

     }
     function HandleLinks(Message1,Message2, Message3)
     {
          var Type=Message3;
          if(Type == "old")
          {
               msg=msg+"<a href='"+Message1+"' target='NetWin'
     onClick=window.open('','NetWin',LinkWindow);>"+Message2+"</a><br>";
          }
          else
          {
               msg="<a href='"+Message1+"' target='NetWin'
     onClick=window.open('','NetWin',LinkWindow);>"+Message2+"</a><br>";
          }
          parent.DataWin.document.Links.msg.value = msg;
          parent.DataWin.document.Links.submit();



     }
     function HandleTime(Message)
```

```
{
                Current_time=Message;
}

function HandleAd(Message1,Message2, Message3)
{
        //Handle the images in the SponsorWin frame
        parent.DataWin.document.Ad.imgsrc.value = Message1;
        parent.DataWin.document.Ad.msg2.value = Message2;
        parent.DataWin.document.Ad.caption.value = Message2;
        parent.DataWin.document.Ad.submit();
}
function WinOpen()
{
        if (CaptionWin == null)
        {
                CaptionWin = window.open("caption2.cfm", "CaptionWin",
"menubar=0,toolbar=0,resizeable=0,width=320,height=360");
                CaptionWin.focus();
                return false;
        }
        else
        {
                if (CaptionWin.closed == true)
                {
                        CaptionWin = window.open("caption2.cfm", "CaptionWin",
"menubar=0,toolbar=0,resizeable=0,width=320,height=360");
                        CaptionWin.focus();
                        return false;
                }
        }
}

function HandleCaption(msg)
{
        if(msg !='')
        {
                if(c_count >5)
                {
                        var first=c_msg.indexOf("||");
                        c_msg = c_msg.substring(first+2);
                        c_msg=c_msg+"\r\n||"+msg;
                }
                else
                {
                        c_msg=c_msg+"\r\n||"+msg;
```

```
                    }
                    //Handle the images in the SponsorWin frame
                    if(CaptionWin == null || CaptionWin.closed == true)
                    {

                    }
                    else
                    {
                            parent.DataWin.document.Caption.caption.value = c_msg;
                            parent.DataWin.document.Caption.submit();
                    }
            c_count++;
            }
    }

    function HandleComments(Message)
    {
            if (IE4)
            {
            document.forms[0].WhatPosition.value=Message;
            }
            else if (Nav4)
            {
            document.forms[0].WhatPosition.value=Message;

            }

    }


    function HandleSessionErrorEx(ErrNumber, ErrDescription)
    {
            alert(ErrDescription);
    }


    function SendElapsedTime()
    {

                    SendElapsedTimeToApplet(Current_time);
                    Current_time+=1000;
                    polling();
    }

    function stopPolling()
    {
```

09951605.050901

```
          clearTimeout(doPolling)
     }
     function polling()
     {

          doPolling = setTimeout("SendElapsedTime()",1000);
     }
     //-->
     </script>

     <CFOUTPUT>
     <body bgcolor="##228B22" leftmargin=3 topmargin=3>
     </CFOUTPUT>

     <applet id="ChatApplet" width="1" height="1" code="ChatClient.class"
     codebase="http://blanche.tcn.net:8090/" VIEWASTEXT mayscript>
     <param name="MessageHandler" value="HandleSessionMessageEx">
     <param name="ElapsedTimeMessageHandler" value="HandleElapsedTime">
     <param name="ErrorHandler" value="HandleSessionErrorEx">
     <param name="SponsorHandler" value="HandleSponsor">
     <param name="LinksHandler" value="HandleLinks">
     <param name="CaptionHandler" value="HandleCaption">
     <param name="AdHandler" value="HandleAd">
     <param name="CommentsHandler" value="HandleComments">
     <param name="CaptionHandler" value="HandleCaption">
     <param name="ErrorHandler" value="HandleSessionErrorEx">
     <param name="TimeHandler" value="HandleTime">
     <CFOUTPUT>
     <param name="ClipSource" value="#ClipSrc#">
     </CFOUTPUT>
     </applet>
     <script Language="JavaScript">
     polling();
     document.applets[0].setStartTime();
     </script>
     <table width="100%" border="0" cellspacing="0" cellpadding="0" align="LEFT"
     valign="TOP">
     <tr>
     <td align="left" valign="top" width="50%">
               <BR>
              </center>
     </td>
     </tr>
     <tr>
     <td colspan="2">
                              <font face="Arial" size="-1">
```

```
                                    <a
href="caption2.cfm?Channel=<cfoutput>#Channel#</cfoutput>" target="CaptionWin"
onClick="return(WinOpen())">CC</a>
</td>
</tr>
</table>
</body>
</html>
</cfif>
```

File: stream4.cfm

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
        <title>Untitled</title>
</head>

<cfif Trim(Channel) is "or channel is ' '>
<body bgcolor="#228B22" leftmargin=3 topmargin=3>

        <a href="http://home.sol.no/~ggunners/NetBird.html">Net Bird</a>

        <br><br><br>
        <table width="80%" align="CENTER" valign="MIDDLE" bgcolor="#F4A460">
        <tr><td>
        <font face="Arial" size="+1" color="Blue">Select a program to view.<br><br>
        Selections button presents archived shows for this demo.<br><br>
        Channels button presents a channel selector mock-up.<br><br>
        Links button shows all the links from the current program.<br><br>
        Almost everything on the screen is a live link to the internet.
</td>
</tr>
        </table>
         &nbsp; <br>


        </center>
        </font>

</body>
</html>
<cfelse>


<script language="javascript">
<!--
var count = 0;
var doPolling;
var Ver4 = parseInt(navigator.appVersion) >= 4;
var Nav4 = ((navigator.appName == "Netscape") && Ver4);
var IE4  = ((navigator.userAgent.indexOf("MSIE") != -1) && Ver4);

var Messages = '';
var msg='';
var c_msg='';
var Current_time=-1000;
CaptionWin=null;
// Save window content dimensions to work around resize
// behaviour in WIN32 versions of Navigator 4.
if (Nav4)
{
        var origWidth = window.innerWidth;
        var origheight = window.innerHeight;
}


// Invoked by <body> tag's onResize event handler to refresh the page for
// Navigator 4 to assist in getting positioned elements back in place.
function RestorePage()
{
        if (Nav4)
        {
                if (origWidth != window.innerWidth || origHeight != window.innerHeight)
                        history.go(0);
        }
}


function SendChatMessage(Author, Type, Message)
{
        document.applets[0].SendMessage(Author, Type, Message);
}

function SendElapsedTimeToApplet(Time)
{
        document.applets[0].SendElapsedTime(Time);
}


function HandleSessionMessageEx(Author, Type, Message)
{
        if (IE4)
        {
        document.forms[0].WhatPosition.value=Message;
        }
        else if (Nav4)
        {
        document.forms[0].WhatPosition.value=Message;
```

```
		}

	}

	function HandleElapsedTime(Message)
	{
		if (IE4)
		{
		document.forms[0].WhatPosition.value=Message;
		}
		else if (Nav4)
		{
		document.forms[0].WhatPosition.value=Message;

		}

	}
	function HandleSponsor(Message1,Message2,Message3)
	{
		//Handle the images in the SponsorWin frame
		parent.DataWin.document.Sponsor.imgsrc.value = Message1;
		parent.DataWin.document.Sponsor.msg2.value = Message2;
		parent.DataWin.document.Sponsor.caption.value = Message2;
		parent.DataWin.document.Sponsor.submit();

	}
	function HandleLinks(Message1,Message2, Message3)
	{
		var Type=Message3;
		if(Type == "old")
		{
			msg=msg+"<a href='"+Message1+"' target='NetWin'
	onClick=window.open('','NetWin',LinkWindow);>"+Message2+"</a><br>";
		}
		else
		{
			msg="<a href='"+Message1+"' target='NetWin'
	onClick=window.open('','NetWin',LinkWindow);>"+Message2+"</a><br>";
		}
		parent.DataWin.document.Links.msg.value = msg;
		parent.DataWin.document.Links.submit();



	}
	function HandleTime(Message)
```

```
{
                Current_time=Message;
}

function HandleAd(Message1,Message2, Message3)
{
        //Handle the images in the SponsorWin frame
        parent.DataWin.document.Ad.imgsrc.value = Message1;
        parent.DataWin.document.Ad.msg2.value = Message2;
        parent.DataWin.document.Ad.caption.value = Message2;
        parent.DataWin.document.Ad.submit();
}
function WinOpen()
{
        if (CaptionWin == null)
        {
                CaptionWin = window.open("caption2.cfm", "CaptionWin",
"menubar=0,toolbar=0,resizeable=0,width=320,height=360");
                CaptionWin.focus();
                return false;
        }
        else
        {
                if (CaptionWin.closed == true)
                {
                        CaptionWin = window.open("caption2.cfm", "CaptionWin",
"menubar=0,toolbar=0,resizeable=0,width=320,height=360");
                        CaptionWin.focus();
                        return false;
                }
        }
}

function HandleCaption(msg)
{
        c_msg=c_msg+"\r\n"+msg;
        //Handle the images in the SponsorWin frame
        if(CaptionWin == null || CaptionWin.closed == true)
        {

        }
        else
        {
                parent.DataWin.document.Caption.caption.value = c_msg;
                parent.DataWin.document.Caption.submit();
        }
}

function HandleComments(Message)
{
        if (IE4)
        {
        document.forms[0].WhatPosition.value=Message;
        }
        else if (Nav4)
        {
        document.forms[0].WhatPosition.value=Message;

        }

}


function HandleSessionErrorEx(ErrNumber, ErrDescription)
{
        alert(ErrDescription);
}


function SendElapsedTime()
{

                SendElapsedTimeToApplet(Current_time);
                Current_time+=1000;
                polling();
}

function stopPolling()
{
     clearTimeout(doPolling)
}
function polling()
{

     doPolling = setTimeout("SendElapsedTime()",1000);
}
//-->
</script>

<CFOUTPUT>
<body bgcolor="##228B22" leftmargin=3 topmargin=3>
</CFOUTPUT>
```

```
<applet id="ChatApplet" width="1" height="1" code="LiveChatClient.class"
codebase="http://blanche.tcn.net:8090/" VIEWASTEXT mayscript>
<param name="MessageHandler" value="HandleSessionMessageEx">
<param name="ElapsedTimeMessageHandler" value="HandleElapsedTime">
<param name="ErrorHandler" value="HandleSessionErrorEx">
<param name="SponsorHandler" value="HandleSponsor">
<param name="LinksHandler" value="HandleLinks">
<param name="CaptionHandler" value="HandleCaption">
<param name="AdHandler" value="HandleAd">
<param name="CommentsHandler" value="HandleComments">
<param name="CaptionHandler" value="HandleCaption">
<param name="ErrorHandler" value="HandleSessionErrorEx">
<param name="TimeHandler" value="HandleTime">
<CFOUTPUT>
<param name="ClipSource" value="#ClipSrc#">
</CFOUTPUT>
</applet>
<script Language="JavaScript">
polling();
document.applets[0].getCurrentPlayTime();
</script>
<table width="100%" border="0" cellspacing="0" cellpadding="0" align="LEFT"
valign="TOP">
<tr>
<td align="left" valign="top">
        <!--- stream in frame name="streamWin" --->

        <cfparam name="Channel" default="">
        <cfparam name="Net" default="">
        <cfparam name="Clipsrc" default="">
        <cfparam name="size" default="">

        <cfif Size is 'S'>
                <cfset Width=160>
                <cfset Height=120>
        <cfelseif Size is 'L'>
                <cfset Width=512>
                <cfset Height=384>
        </cfif>

        <cfif Not Clipsrc is ">
                <cfif ClipSrc is "SunPar3" and Size is "L">
                        <cfset cliploc =
 'http://www.broadcast.com/business/sun/98par3challenge_102099_100.ram'>
                <cfelseif ClipSrc is "SunPar3" and Size is "S">
```

09851605.050901

```
                    <cfset cliploc =
'http://www.broadcast.com/business/sun/98par3challenge_102099_56.ram'>
               <cfelse>
                    <cfif Size is 'S'>
                         <cfset cliploc = '/media/56k/#Clipsrc#.ram'>
                    <cfelse>
                         <cfset cliploc = '/media/#Clipsrc#.ram'>
                    </cfif>
               </cfif>
          </cfif>


          <center>
          <cfoutput>
          <object
           ID=video1
           CLASSID="clsid:CFCDAA03-8BE4-11cf-B84B-0020AFBBCCFA"
           height=#Height#
           width=#Width#
           >

           <param name="controls" value="ImageWindow">
           <param NAME="console" VALUE="Clip1">
           <param NAME="autostart" VALUE="true">
           <param NAME="src" VALUE="#Cliploc#">
           <embed name="video1" SRC="#Cliploc#"
           type="audio/x-pn-realaudio-plugin"
           CONSOLE="Clip1"
           controls="ImageWindow"
           height=#Height#
           width=#Width#
           AUTOSTART=true>
           </object>

           <BR>
           <form>
           <input type="Button" name="Play_Button" value="Play"
onClick="document.applets[0].init();document.applets[0].getCurrentPlayTime();document.vi
deo1.DoPlay();polling()">
           <input type="Button" name="Pause_Button" value="Pause"
onClick="document.video1.DoPause();stopPolling();">
           <input type="Button" name="Stop_Button" value="Stop"
onClick="document.video1.DoStop();stopPolling()">
<!---      <input type="Button" name="position" value="Position"
onClick="javascript:alert(document.video1.GetPosition())">
```

```
          <input type="Button" name="buffer" value="Send"
onClick="SendChatMessage('ripu', 'ChatMessage', document.video1.GetPosition())">
          <input type="Text" name="WhatPosition"> --->
         </form>
          </cfoutput>
         </center>
</td>
</tr>
<tr>
<td colspan="2">
                                       <font face="Arial" size="-1">
                                                <a
href="caption2.cfm?Channel=<cfoutput>#Channel#</cfoutput>" target="CaptionWin"
onClick="return(WinOpen())">CC</a>
</td>
</tr>
</table>
</body>
</html>
</cfif>


<!--- <table width="100%" border="0" cellspacing="0" cellpadding="0" align="LEFT"
valign="TOP">
<tr>
<td align="left" valign="top" width="50%">
          <BR>
         </center>
</td>
</tr>
<tr>
<td colspan="2">
                                       <font face="Arial" size="-1">
                                                <a
href="caption2.cfm?Channel=<cfoutput>#Channel#</cfoutput>" target="CaptionWin"
onClick="return(WinOpen())">CC</a>
</td>
</tr>
</table>
</body>
</html>
</cfif> --->
```

File: thankyou.cfm

```
<cfquery name="getcomments"
                datasource="VBIContent"
                username="caption"
                password="trewq55">

select comment from ctlfile
where clipsource = '#clipsrc#'
and Type = 'C'
and seconds > #Evaluate(LTime -10)#
and seconds < #Evaluate(LTime +10)#
order by seconds


</cfquery>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
        <title>Thank You</title>
</head>

<body>
<font face="Arial" color = "cadetblue" size="-1">Thank you for your comment. Here are the
comments around yours:
<font color="crimson">
<br>
<cfoutput query="GetComments">
#comment#<br>
</cfoutput></font>
</font>
<form>
<input type="Button" value="Close" onClick="window.close();">
</form>
</body>
</html>
```

Appendix D

Copyright © 2000 I LOVE TV INC. (868970 Ontario Inc.)
All rights reserved unless otherwise noted

Cold Fusion

TESsocket

File: captionsock.c

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <fcntl.h>
#include <errno.h>

#define EOCAP (buf[0]==0x01 || buf[0]==0x02 || buf[0]==0x11 || buf[0]==0x12)
#define LOG(X) {fprintf(fp,"\n%s",X); fflush(fp);}
#define DEBUG 0

#define captionsock 9000
#define IPTES3 "206.221.245.75"

FILE *fp;

#define stderr fp

static parse();

int argc;
char **argv;
int state;
char outstr[10000];
int sock_desc;                /* Socket descriptor */
struct sockaddr_in sa;        /* Internet Socket addr. struct */
struct hostent *hp;           /* Result of host name lookup  */
struct servent *sp;           /* Result of service lookup */
int len,seq,i,flags=0;
char seq_str[30];
char rep_type;
char query_str[90000];
FILE *fp_head,*fp_list,*fp_tail;
```

```
char ch;
char ARGV[26][9000];
int ARGC;

int bargc;
char *bargv[20],*cp;

char myname[] = "captionsock";              /* this program nanme */

int inc;
char  junk[9000];
char buf[9000];


/* all remaining test passed to program is then passed
   directly to upstream */

int count;
int lenr,res;
char blank[]="";


#define MAXHOSTNAME 32
#define BUFSIZE 9000
int
main ( )
{

char sendbuff[300]; /* due to captioning data size limitiations,
                              this should be more than enough */
int buffin;
char *buffpnt;
int firsttime=1;
char field[2];
char outb[1000];
char systemstring[200];

char localhost[32];

  fp=fopen("/tmp/captionlog","w");
/* caption log stores raw data!!! */

  /*******************************************************/
  /********************************************************/


  gethostname(localhost, MAXHOSTNAME);
```

```
if((hp = gethostbyname(localhost)) == NULL) {
        fprintf(stderr,"no such host?\n");
        exit(0);
}

sa.sin_port = htons(captionsock);
bcopy((char *)hp->h_addr, (char *)&sa.sin_addr, hp->h_length);
sa.sin_family = hp->h_addrtype;
inet_aton(IPTES3,&sa.sin_addr);
/* IP of TES3!! */

if((sock_desc=socket(hp->h_addrtype,SOCK_STREAM,0)) < 0) {
        perror("socket");
        exit(0);
}


LOG("CONNECT SOCKET");
if(connect(sock_desc,&sa,(int)(sizeof sa)) < 0) {
        perror("connect");
        exit(0);
}

LOG("READ");
/* send these lines off to mysql as they come in */
/* insert into record incrementing table, with time stamp on creation */
/* perl program will collect together fragmented time stamped data,
  and join to completed objects ... then on from there*/

/* TOKENS at this time are  0x01 0x02 0x11 0x12 0x0a */

field[1]='\0';

buffpnt=sendbuff;

/* on first in, discard everything up to first valid TOKEN */

for (;;) {
  len = read(sock_desc,buf,1);
  buf[len]='\0';
  buf[0]=buf[0]&0x7f; /* mask of high bit */
/* DEBUG:  printf("%s",buf);  */

  if( len > 0 )
    {
        if( EOCAP )
          {
               *buffpnt='\0';
               if( firsttime ) { firsttime=0; buffpnt=sendbuff;}
               else
                 {
                      /* set up shell cmd line to insert to mysql!! */
                   systemstring[0]='\0';
                      strcat(systemstring,"echo 'insert into raw (str2,stamped) values (\"");
                      strcat(systemstring, field);

                      /* loop over data and make comma delimited*/
                      strcat(systemstring,  " ");

                      for(i=0; i<strlen(sendbuff); i++)
                        {
                         sprintf(outb,"%d",(int)sendbuff[i]);
                         strcat(systemstring, outb);
                         strcat(systemstring, ",");
                        }

    /* echo 'insert into raw (str2,stamped) values ("2 34,23,",now())' | /usr/www/local/mysql-
    3.22.27/bin/mysql caption */

    /* strcat(systemstring, "\",now())' >> /tmp/yyyy");        */


                        strcat(systemstring, "\",now())'  | /usr/www/local/mysql-3.22.27/bin/mysql
    caption");


    /* comment out logic below to send data if also blank */
                     if( strlen(sendbuff) > 0 )
                            system(systemstring);

                        buffpnt=sendbuff;
                  } /* end else */

                 switch ((char)(buf[0]))
                  {
                            case 0x01: field[0]='1'; break;
                            case 0x02: field[0]='2'; break;
                            case 0x11: field[0]='1'; break;
                            case 0x12: field[0]='2'; break;
                            default: exit(0); break;
                  }
```

```
      } /* end if EOCAP */
     else  /* not end of caption char */
      {
       /* if(buf[0]==0x0a ) { buf[0]=0xff; exit(0); } */
          *(buffpnt)=buf[0]; buffpnt++;
        LOG(buf);
      }

    } /* end len > 0 */

    /* SYSTEM shell call -  although inefficent right know, this still can write over 100 sub
string events to the DB
per second, far beyond what we need!!! */

    /* system("date  >> /tmp/yy"); */
   }

/* NOTE:  01 02 11 12 (hex)  01 02  start of data   11 12 no data
0x0a just means nothing ? */


}
```

File: cgic.c

```
#if CGICDEBUG
#define CGICDEBUGSTART \
        { \
                FILE *dout; \
                dout = fopen("/home/boutell/public_html/debug", "a"); \

#define CGICDEBUGEND \
                fclose(dout); \
        }
#else /* CGICDEBUG */
#define CGICDEBUGSTART
#define CGICDEBUGEND
#endif /* CGICDEBUG */

#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <stdlib.h>
#ifndef NO_UNISTD
#include <unistd.h>
#endif /* NO_UNISTD */
#include "/usr/local/cgic103/cgic.h"

#define cgiStrEq(a, b) (!strcmp((a), (b)))

char *cgiServerSoftware;
char *cgiServerName;
char *cgiGatewayInterface;
char *cgiServerProtocol;
char *cgiServerPort;
char *cgiRequestMethod;
char *cgiPathInfo;
char *cgiPathTranslated;
char *cgiScriptName;
char *cgiQueryString;
char *cgiRemoteHost;
char *cgiRemoteAddr;
char *cgiAuthType;
char *cgiRemoteUser;
char *cgiRemoteIdent;
char *cgiContentType;
int cgiContentLength;
char *cgiAccept;
char *cgiUserAgent;

FILE *cgiIn;
FILE *cgiOut;

/* One form entry, consisting of an attribute-value pair. */

typedef struct cgiFormEntryStruct {
      char *attr;
      char *value;
      struct cgiFormEntryStruct *next;
} cgiFormEntry;

/* The first form entry. */
static cgiFormEntry *cgiFormEntryFirst;

/* True if CGI environment was restored from a file. */
static int cgiRestored = 0;

static void cgiGetenv(char **s, char *var);

typedef enum {
        cgiParseSuccess,
        cgiParseMemory,
        cgiParseIO
} cgiParseResultType;

static cgiParseResultType cgiParseGetFormInput();
static cgiParseResultType cgiParsePostFormInput();
static cgiParseResultType cgiParseFormInput(char *data, int length);
static void cgiSetupConstants();
static void cgiFreeResources();
static int cgiStrEqNc(char *s1, char *s2);

int main(int argc, char *argv[]) {
        int result;
        char *cgiContentLengthString;
        cgiSetupConstants();
        cgiGetenv(&cgiServerSoftware, "SERVER_SOFTWARE");
        cgiGetenv(&cgiServerName, "SERVER_NAME");
        cgiGetenv(&cgiGatewayInterface, "GATEWAY_INTERFACE");
        cgiGetenv(&cgiServerProtocol, "SERVER_PROTOCOL");
        cgiGetenv(&cgiServerPort, "SERVER_PORT");
        cgiGetenv(&cgiRequestMethod, "REQUEST_METHOD");
        cgiGetenv(&cgiPathInfo, "PATH_INFO");
        cgiGetenv(&cgiPathTranslated, "PATH_TRANSLATED");
```

```
          cgiGetenv(&cgiScriptName, "SCRIPT_NAME");
          cgiGetenv(&cgiQueryString, "QUERY_STRING");
          cgiGetenv(&cgiRemoteHost, "REMOTE_HOST");
          cgiGetenv(&cgiRemoteAddr, "REMOTE_ADDR");
          cgiGetenv(&cgiAuthType, "AUTH_TYPE");
          cgiGetenv(&cgiRemoteUser, "REMOTE_USER");
          cgiGetenv(&cgiRemoteIdent, "REMOTE_IDENT");
          cgiGetenv(&cgiContentType, "CONTENT_TYPE");
          cgiGetenv(&cgiContentLengthString, "CONTENT_LENGTH");
          cgiContentLength = atoi(cgiContentLengthString);
          cgiGetenv(&cgiAccept, "HTTP_ACCEPT");
          cgiGetenv(&cgiUserAgent, "HTTP_USER_AGENT");
#ifdef CGICDEBUG
          CGICDEBUGSTART
          fprintf(dout, "%d\n", cgiContentLength);
          fprintf(dout, "%s\n", cgiRequestMethod);
          fprintf(dout, "%s\n", cgiContentType);
          CGICDEBUGEND
#endif /* CGICDEBUG */
          cgiFormEntryFirst = 0;
          cgiIn = stdin;
          cgiOut = stdout;
          cgiRestored = 0;


          /* These five lines keep compilers from
                    producing warnings that argc and argv
                    are unused. They have no actual function. */
          if (argc) {
                    if (argv[0]) {
                              cgiRestored = 0;
                    }
          }


          if (cgiStrEqNc(cgiRequestMethod, "post")) {
#ifdef CGICDEBUG
                    CGICDEBUGSTART
                    fprintf(dout, "POST recognized\n");
                    CGICDEBUGEND
#endif /* CGICDEBUG */
                    if (cgiStrEqNc(cgiContentType, "application/x-www-form-urlencoded")) {

#ifdef CGICDEBUG
                              CGICDEBUGSTART
                              fprintf(dout, "Calling PostFormInput\n");
```

```
            CGICDEBUGEND
#endif /* CGICDEBUG */
            if (cgiParsePostFormInput() != cgiParseSuccess) {
#ifdef CGICDEBUG
                CGICDEBUGSTART
                fprintf(dout, "PostFormInput failed\n");
                CGICDEBUGEND
#endif /* CGICDEBUG */
                cgiFreeResources();
                return -1;
            }
#ifdef CGICDEBUG
            CGICDEBUGSTART
            fprintf(dout, "PostFormInput succeeded\n");
            CGICDEBUGEND
#endif /* CGICDEBUG */
        }
    } else if (cgiStrEqNc(cgiRequestMethod, "get")) {
        /* The spec says this should be taken care of by
            the server, but... it isn't */
        cgiContentLength = strlen(cgiQueryString);
        if (cgiParseGetFormInput() != cgiParseSuccess) {
#ifdef CGICDEBUG
            CGICDEBUGSTART
            fprintf(dout, "GetFormInput failed\n");
            CGICDEBUGEND
#endif /* CGICDEBUG */
            cgiFreeResources();
            return -1;
        } else {
#ifdef CGICDEBUG
            CGICDEBUGSTART
            fprintf(dout, "GetFormInput succeeded\n");
            CGICDEBUGEND
#endif /* CGICDEBUG */
        }
    }
    result = cgiMain();
    cgiFreeResources();

    /* The following two lines allegedly produce better behavior
        when run with the CERN server on some platforms.
        Remove them and experiment if you wish. */
    fflush(stdout);
    sleep(1);
    return result;
```

```
}

static void cgiGetenv(char **s, char *var){
        *s = getenv(var);
        if (!(*s)) {
                *s = "";
        }
}

static cgiParseResultType cgiParsePostFormInput() {
        char *input;
        cgiParseResultType result;
        if (!cgiContentLength) {
                return cgiParseSuccess;
        }
        input = (char *) malloc(cgiContentLength);
        if (!input) {
                return cgiParseMemory;
        }
        if (fread(input, 1, cgiContentLength, cgiIn) != cgiContentLength) {
                return cgiParseIO;
        }
        result = cgiParseFormInput(input, cgiContentLength);
        free(input);
        return result;
}

static cgiParseResultType cgiParseGetFormInput() {
        return cgiParseFormInput(cgiQueryString, cgiContentLength);
}

typedef enum {
        cgiEscapeRest,
        cgiEscapeFirst,
        cgiEscapeSecond
} cgiEscapeState;

typedef enum {
        cgiUnescapeSuccess,
        cgiUnescapeMemory
} cgiUnescapeResultType;

static cgiUnescapeResultType cgiUnescapeChars(char **sp, char *cp, int len);

static cgiParseResultType cgiParseFormInput(char *data, int length) {
        /* Scan for pairs, unescaping and storing them as they are found. */
```

```
int pos = 0;
cgiFormEntry *n;
cgiFormEntry *l = 0;
while (pos != length) {
        int foundEq = 0;
        int foundAmp = 0;
        int start = pos;
        int len = 0;
        char *attr;
        char *value;
        while (pos != length) {
                if (data[pos] == '=') {
                        foundEq = 1;
                        pos++;
                        break;
                }
                pos++;
                len++;
        }
        if (!foundEq) {
                break;
        }
        if (cgiUnescapeChars(&attr, data+start, len)
                != cgiUnescapeSuccess) {
                return cgiParseMemory;
        }
        start = pos;
        len = 0;
        while (pos != length) {
                if (data[pos] == '&') {
                        foundAmp = 1;
                        pos++;
                        break;
                }
                pos++;
                len++;
        }
        /* The last pair probably won't be followed by a &, but
                that's fine, so check for that after accepting it */
        if (cgiUnescapeChars(&value, data+start, len)
                != cgiUnescapeSuccess) {
                return cgiParseMemory;
        }
        /* OK, we have a new pair, add it to the list. */
        n = (cgiFormEntry *) malloc(sizeof(cgiFormEntry));
        if (!n) {
```

```
				return cgiParseMemory;
			}
			n->attr = attr;
			n->value = value;
			n->next = 0;
			if (!l) {
				cgiFormEntryFirst = n;
			} else {
				l->next = n;
			}
			l = n;
			if (!foundAmp) {
				break;
			}
		}
		return cgiParseSuccess;
}

static int cgiHexValue[256];

cgiUnescapeResultType cgiUnescapeChars(char **sp, char *cp, int len) {
		char *s;
		cgiEscapeState escapeState = cgiEscapeRest;
		int escapedValue = 0;
		int srcPos = 0;
		int dstPos = 0;
		s = (char *) malloc(len + 1);
		if (!s) {
			return cgiUnescapeMemory;
		}
		while (srcPos < len) {
			int ch = cp[srcPos];
			switch (escapeState) {
				case cgiEscapeRest:
				if (ch == '%') {
					escapeState = cgiEscapeFirst;
				} else if (ch == '+') {
					s[dstPos++] = ' ';
				} else {
					s[dstPos++] = ch;
				}
				break;
				case cgiEscapeFirst:
				escapedValue = cgiHexValue[ch] << 4;
				escapeState = cgiEscapeSecond;
				break;
```

```
            case cgiEscapeSecond:
            escapedValue += cgiHexValue[ch];
            s[dstPos++] = escapedValue;
            escapeState = cgiEscapeRest;
            break;
        }
        srcPos++;
    }
    s[dstPos] = '\0';
    *sp = s;
    return cgiUnescapeSuccess;
}

static void cgiSetupConstants() {
    int i;
    for (i=0; (i < 256); i++) {
        cgiHexValue[i] = 0;
    }
    cgiHexValue['0'] = 0;
    cgiHexValue['1'] = 1;
    cgiHexValue['2'] = 2;
    cgiHexValue['3'] = 3;
    cgiHexValue['4'] = 4;
    cgiHexValue['5'] = 5;
    cgiHexValue['6'] = 6;
    cgiHexValue['7'] = 7;
    cgiHexValue['8'] = 8;
    cgiHexValue['9'] = 9;
    cgiHexValue['A'] = 10;
    cgiHexValue['B'] = 11;
    cgiHexValue['C'] = 12;
    cgiHexValue['D'] = 13;
    cgiHexValue['E'] = 14;
    cgiHexValue['F'] = 15;
    cgiHexValue['a'] = 10;
    cgiHexValue['b'] = 11;
    cgiHexValue['c'] = 12;
    cgiHexValue['d'] = 13;
    cgiHexValue['e'] = 14;
    cgiHexValue['f'] = 15;
}

static void cgiFreeResources() {
    cgiFormEntry *c = cgiFormEntryFirst;
    cgiFormEntry *n;
    while (c) {
```

```
                n = c->next;
                free(c->attr);
                free(c->value);
                free(c);
                c = n;
        }
        /* If the cgi environment was restored from a saved environment,
                then these are in allocated space and must also be freed */
        if (cgiRestored) {
                free(cgiServerSoftware);
                free(cgiServerName);
                free(cgiGatewayInterface);
                free(cgiServerProtocol);
                free(cgiServerPort);
                free(cgiRequestMethod);
                free(cgiPathInfo);
                free(cgiPathTranslated);
                free(cgiScriptName);
                free(cgiQueryString);
                free(cgiRemoteHost);
                free(cgiRemoteAddr);
                free(cgiAuthType);
                free(cgiRemoteUser);
                free(cgiRemoteIdent);
                free(cgiContentType);
                free(cgiAccept);
                free(cgiUserAgent);
        }
}

static cgiFormResultType cgiFormEntryString(
        cgiFormEntry *e, char *result, int max, int newlines);

static cgiFormEntry *cgiFormEntryFindFirst(char *name);
static cgiFormEntry *cgiFormEntryFindNext();

cgiFormResultType cgiFormString(
      char *name, char *result, int max) {
        cgiFormEntry *e;
        e = cgiFormEntryFindFirst(name);
        if (!e) {
                strcpy(result, "");
                return cgiFormNotFound;
        }
        return cgiFormEntryString(e, result, max, 1);
}
```

```
cgiFormResultType cgiFormStringNoNewlines(
      char *name, char *result, int max) {
         cgiFormEntry *e;
         e = cgiFormEntryFindFirst(name);
         if (!e) {
                  strcpy(result, "");
                  return cgiFormNotFound;
         }
         return cgiFormEntryString(e, result, max, 0);
}

cgiFormResultType cgiFormStringMultiple(
      char *name, char ***result) {
         char **stringArray;
         cgiFormEntry *e;
         int i;
         int total = 0;
         /* Make two passes. One would be more efficient, but this
                  function is not commonly used. The select menu and
                  radio box functions are faster. */
         e = cgiFormEntryFindFirst(name);
         if (e != 0) {
                  do {
                           total++;
                  } while ((e = cgiFormEntryFindNext()) != 0);
         }
         stringArray = (char **) malloc(sizeof(char *) * (total + 1));
         if (!stringArray) {
                  *result = 0;
                  return cgiFormMemory;
         }
         /* initialize all entries to null; the last will stay that way */
         for (i=0; (i <= total); i++) {
                  stringArray[i] = 0;
         }
         /* Now go get the entries */
         e = cgiFormEntryFindFirst(name);
#ifdef CGICDEBUG
         CGICDEBUGSTART
         fprintf(dout, "StringMultiple Beginning\n");
         CGICDEBUGEND
#endif /* CGICDEBUG */
         if (e) {
                  i = 0;
                  do {
                        int max = strlen(e->value) + 1;
                        stringArray[i] = (char *) malloc(max);
                        if (stringArray[i] == 0) {
                                /* Memory problems */
                                cgiStringArrayFree(stringArray);
                                *result = 0;
                                return cgiFormMemory;
                        }
                        strcpy(stringArray[i], e->value);
                        cgiFormEntryString(e, stringArray[i], max, 1);
                        i++;
                } while ((e = cgiFormEntryFindNext()) != 0);
                *result = stringArray;
#ifdef CGICDEBUG
                CGICDEBUGSTART
                fprintf(dout, "StringMultiple Succeeding\n");
                CGICDEBUGEND
#endif /* CGICDEBUG */
                return cgiFormSuccess;
        } else {
                *result = stringArray;
#ifdef CGICDEBUG
                CGICDEBUGSTART
                fprintf(dout, "StringMultiple found nothing\n");
                CGICDEBUGEND
#endif /* CGICDEBUG */
                return cgiFormNotFound;
        }
}

cgiFormResultType cgiFormStringSpaceNeeded(
        char *name, int *result) {
        cgiFormEntry *e;
        e = cgiFormEntryFindFirst(name);
        if (!e) {
                *result = 1;
                return cgiFormNotFound;
        }
        *result = strlen(e->value) + 1;
        return cgiFormSuccess;
}

static cgiFormResultType cgiFormEntryString(
        cgiFormEntry *e, char *result, int max, int newlines) {
        char *dp, *sp;
        int truncated = 0;
int len = 0;
int avail = max-1;
int crCount = 0;
int lfCount = 0;
dp = result;
sp = e->value;
while (1) {
    int ch;
    ch = *sp;
    if (len == avail) {
        truncated = 1;
        break;
    }
    /* Fix the CR/LF, LF, CR nightmare: watch for
        consecutive bursts of CRs and LFs in whatever
        pattern, then actually output the larger number
        of LFs. Consistently sane, yet it still allows
        consecutive blank lines when the user
        actually intends them. */
    if ((ch == 13) || (ch == 10)) {
        if (ch == 13) {
            crCount++;
        } else {
            lfCount++;
        }
    } else {
        if (crCount || lfCount) {
            int lfsAdd = crCount;
            if (lfCount > crCount) {
                lfsAdd = lfCount;
            }
            /* Stomp all newlines if desired */
            if (!newlines) {
                lfsAdd = 0;
            }
            while (lfsAdd) {
                if (len == avail) {
                    truncated = 1;
                    break;
                }
                *dp = 10;
                dp++;
                lfsAdd--;
                len++;
            }
            crCount = 0;
```

```
                        lfCount = 0;
                    }
                    if (ch == '\0') {
                        /* The end of the source string */
                        break;
                    }
                    *dp = ch;
                    dp++;
                    len++;
                }
                sp++;
            }
            *dp = '\0';
            if (truncated) {
                return cgiFormTruncated;
            } else if (!len) {
                return cgiFormEmpty;
            } else {
                return cgiFormSuccess;
            }
        }

static int cgiFirstNonspaceChar(char *s);

cgiFormResultType cgiFormInteger(
        char *name, int *result, int defaultV) {
    cgiFormEntry *e;
    int ch;
    e = cgiFormEntryFindFirst(name);
    if (!e) {
        *result = defaultV;
        return cgiFormNotFound;
    }
    if (!strlen(e->value)) {
        *result = defaultV;
        return cgiFormEmpty;
    }
    ch = cgiFirstNonspaceChar(e->value);
    if (!(isdigit(ch)) && (ch != '-') && (ch != '+')) {
        *result = defaultV;
        return cgiFormBadType;
    } else {
        *result = atoi(e->value);
        return cgiFormSuccess;
    }
}
```

09554505.050604

```
cgiFormResultType cgiFormIntegerBounded(
    char *name, int *result, int min, int max, int defaultV) {
      cgiFormResultType error = cgiFormInteger(name, result, defaultV);
      if (error != cgiFormSuccess) {
            return error;
      }
      if (*result < min) {
            *result = min;
            return cgiFormConstrained;
      }
      if (*result > max) {
            *result = max;
            return cgiFormConstrained;
      }
      return cgiFormSuccess;
}

cgiFormResultType cgiFormDouble(
    char *name, double *result, double defaultV) {
      cgiFormEntry *e;
      int ch;
      e = cgiFormEntryFindFirst(name);
      if (!e) {
            *result = defaultV;
            return cgiFormNotFound;
      }
      if (!strlen(e->value)) {
            *result = defaultV;
            return cgiFormEmpty;
      }
      ch = cgiFirstNonspaceChar(e->value);
      if (!(isdigit(ch)) && (ch != '.') && (ch != '-') && (ch != '+')) {
            *result = defaultV;
            return cgiFormBadType;
      } else {
            *result = atof(e->value);
            return cgiFormSuccess;
      }
}

cgiFormResultType cgiFormDoubleBounded(
    char *name, double *result, double min, double max, double defaultV) {
      cgiFormResultType error = cgiFormDouble(name, result, defaultV);
      if (error != cgiFormSuccess) {
            return error;
```

```
		}
		if (*result < min) {
			*result = min;
			return cgiFormConstrained;
		}
		if (*result > max) {
			*result = max;
			return cgiFormConstrained;
		}
		return cgiFormSuccess;
	}

	cgiFormResultType cgiFormSelectSingle(
		char *name, char **choicesText, int choicesTotal,
		int *result, int defaultV)
	{
		cgiFormEntry *e;
		int i;
		e = cgiFormEntryFindFirst(name);
	#ifdef CGICDEBUG
		CGICDEBUGSTART
		fprintf(dout, "%d\n", (int) e);
		CGICDEBUGEND
	#endif /* CGICDEBUG */
		if (!e) {
			*result = defaultV;
			return cgiFormNotFound;
		}
		for (i=0; (i < choicesTotal); i++) {
	#ifdef CGICDEBUG
			CGICDEBUGSTART
			fprintf(dout, "%s %s\n", choicesText[i], e->value);
			CGICDEBUGEND
	#endif /* CGICDEBUG */
			if (cgiStrEq(choicesText[i], e->value)) {
	#ifdef CGICDEBUG
				CGICDEBUGSTART
				fprintf(dout, "MATCH\n");
				CGICDEBUGEND
	#endif /* CGICDEBUG */
				*result = i;
				return cgiFormSuccess;
			}
		}
		*result = defaultV;
		return cgiFormNoSuchChoice;
```

```
}

cgiFormResultType cgiFormSelectMultiple(
	char *name, char **choicesText, int choicesTotal,
	int *result, int *invalid)
{
	cgiFormEntry *e;
	int i;
	int hits = 0;
	int invalidE = 0;
	for (i=0; (i < choicesTotal); i++) {
		result[i] = 0;
	}
	e = cgiFormEntryFindFirst(name);
	if (!e) {
		*invalid = invalidE;
		return cgiFormNotFound;
	}
	do {
		int hit = 0;
		for (i=0; (i < choicesTotal); i++) {
			if (cgiStrEq(choicesText[i], e->value)) {
				result[i] = 1;
				hits++;
				hit = 1;
				break;
			}
		}
		if (!(hit)) {
			invalidE++;
		}
	} while ((e = cgiFormEntryFindNext()) != 0);

	*invalid = invalidE;

	if (hits) {
		return cgiFormSuccess;
	} else {
		return cgiFormNotFound;
	}
}

cgiFormResultType cgiFormCheckboxSingle(
	char *name)
{
	cgiFormEntry *e;
```

09514505.050904

```
	e = cgiFormEntryFindFirst(name);
	if (!e) {
		return cgiFormNotFound;
	}
	return cgiFormSuccess;
}

extern cgiFormResultType cgiFormCheckboxMultiple(
	char *name, char **valuesText, int valuesTotal,
	int *result, int *invalid)
{
	/* Implementation is identical to cgiFormSelectMultiple. */
	return cgiFormSelectMultiple(name, valuesText,
		valuesTotal, result, invalid);
}

cgiFormResultType cgiFormRadio(
	char *name,
	char **valuesText, int valuesTotal, int *result, int defaultV)
{
	/* Implementation is identical to cgiFormSelectSingle. */
	return cgiFormSelectSingle(name, valuesText, valuesTotal,
		result, defaultV);
}

void cgiHeaderLocation(char *redirectUrl) {
	fprintf(cgiOut, "Location: %s%c%c", redirectUrl, 10, 10);
}

void cgiHeaderStatus(int status, char *statusMessage) {
	fprintf(cgiOut, "Status: %d %s%c%c", status, statusMessage,
	10, 10);
}

void cgiHeaderContentType(char *mimeType) {
	fprintf(cgiOut, "Content-type: %s%c%c", mimeType, 10, 10);
}

#ifndef NO_SYSTEM

int cgiSaferSystem(char *command) {
	char *s;
	char *sp;
	int i;
	int len = (strlen(command) * 2) + 1;
	s = (char *) malloc(len);
```

```
		if (!s) {
			return -1;
		}
		sp = s;
		for (i=0; (i < len); i++) {
			if (command[i] == ';') {
				*sp = '\\';
				sp++;
			} else if (command[i] == '|') {
				*sp = '\\';
				sp++;
			}
			*sp = command[i];
			sp++;
		}
		*sp = '\0';
		return system(s);
	}

	#endif /* NO_SYSTEM */

	static int cgiWriteString(FILE *out, char *s);

	static int cgiWriteInt(FILE *out, int i);

	cgiEnvironmentResultType cgiWriteEnvironment(char *filename) {
		FILE *out;
		cgiFormEntry *e;
		/* Be sure to open in binary mode */
		out = fopen(filename, "wb");
		if (!out) {
			/* Can't create file */
			return cgiEnvironmentIO;
		}
		if (!cgiWriteString(out, cgiServerSoftware)) {
			goto error;
		}
		if (!cgiWriteString(out, cgiServerName)) {
			goto error;
		}
		if (!cgiWriteString(out, cgiGatewayInterface)) {
			goto error;
		}
		if (!cgiWriteString(out, cgiServerProtocol)) {
			goto error;
		}
```

```
if (!cgiWriteString(out, cgiServerPort)) {
        goto error;
}
if (!cgiWriteString(out, cgiRequestMethod)) {
        goto error;
}
if (!cgiWriteString(out, cgiPathInfo)) {
        goto error;
}
if (!cgiWriteString(out, cgiPathTranslated)) {
        goto error;
}
if (!cgiWriteString(out, cgiScriptName)) {
        goto error;
}
if (!cgiWriteString(out, cgiQueryString)) {
        goto error;
}
if (!cgiWriteString(out, cgiRemoteHost)) {
        goto error;
}
if (!cgiWriteString(out, cgiRemoteAddr)) {
        goto error;
}
if (!cgiWriteString(out, cgiAuthType)) {
        goto error;
}
if (!cgiWriteString(out, cgiRemoteUser)) {
        goto error;
}
if (!cgiWriteString(out, cgiRemoteIdent)) {
        goto error;
}
if (!cgiWriteString(out, cgiContentType)) {
        goto error;
}
if (!cgiWriteString(out, cgiAccept)) {
        goto error;
}
if (!cgiWriteString(out, cgiUserAgent)) {
        goto error;
}
if (!cgiWriteInt(out, cgiContentLength)) {
        goto error;
}
e = cgiFormEntryFirst;
```

```
		while (e) {
			if (!cgiWriteString(out, e->attr)) {
				goto error;
			}
			if (!cgiWriteString(out, e->value)) {
				goto error;
			}
			e = e->next;
		}
		fclose(out);
		return cgiEnvironmentSuccess;
	error:
		fclose(out);
		/* If this function is not defined in your system,
			you must substitute the appropriate
			file-deletion function. */
		unlink(filename);
		return cgiEnvironmentIO;
	}

	static int cgiWriteString(FILE *out, char *s) {
		int len = strlen(s);
		cgiWriteInt(out, len);
		if (fwrite(s, 1, len, out) != len) {
			return 0;
		}
		return 1;
	}

	static int cgiWriteInt(FILE *out, int i) {
		if (!fwrite(&i, sizeof(int), 1, out)) {
			return 0;
		}
		return 1;
	}

	static int cgiReadString(FILE *out, char **s);

	static int cgiReadInt(FILE *out, int *i);

	cgiEnvironmentResultType cgiReadEnvironment(char *filename) {
		FILE *in;
		cgiFormEntry *e, *p;
		/* Free any existing data first */
		cgiFreeResources();
		/* Be sure to open in binary mode */
```

```
in = fopen(filename, "rb");
if (!in) {
        /* Can't access file */
        return cgiEnvironmentIO;
}
if (!cgiReadString(in, &cgiServerSoftware)) {
        goto error;
}
if (!cgiReadString(in, &cgiServerName)) {
        goto error;
}
if (!cgiReadString(in, &cgiGatewayInterface)) {
        goto error;
}
if (!cgiReadString(in, &cgiServerProtocol)) {
        goto error;
}
if (!cgiReadString(in, &cgiServerPort)) {
        goto error;
}
if (!cgiReadString(in, &cgiRequestMethod)) {
        goto error;
}
if (!cgiReadString(in, &cgiPathInfo)) {
        goto error;
}
if (!cgiReadString(in, &cgiPathTranslated)) {
        goto error;
}
if (!cgiReadString(in, &cgiScriptName)) {
        goto error;
}
if (!cgiReadString(in, &cgiQueryString)) {
        goto error;
}
if (!cgiReadString(in, &cgiRemoteHost)) {
        goto error;
}
if (!cgiReadString(in, &cgiRemoteAddr)) {
        goto error;
}
if (!cgiReadString(in, &cgiAuthType)) {
        goto error;
}
if (!cgiReadString(in, &cgiRemoteUser)) {
        goto error;
```

```
	}
	if (!cgiReadString(in, &cgiRemoteIdent)) {
		goto error;
	}
	if (!cgiReadString(in, &cgiContentType)) {
		goto error;
	}
	if (!cgiReadString(in, &cgiAccept)) {
		goto error;
	}
	if (!cgiReadString(in, &cgiUserAgent)) {
		goto error;
	}
	if (!cgiReadInt(in, &cgiContentLength)) {
		goto error;
	}
	p = 0;
	while (1) {
		e = (cgiFormEntry *) malloc(sizeof(cgiFormEntry));
		if (!e) {
			cgiFreeResources();
			fclose(in);
			return cgiEnvironmentMemory;
		}
		if (!cgiReadString(in, &e->attr)) {
			/* This means we've reached the end of the list. */
			free(e);
			break;
		}
		if (!cgiReadString(in, &e->value)) {
			free(e);
			goto error;
		}
		e->next = 0;
		if (p) {
			p->next = e;
		} else {
			cgiFormEntryFirst = e;
		}
		p = e;
	}
	fclose(in);
	cgiRestored = 1;
	return cgiEnvironmentSuccess;
error:
	cgiFreeResources();
```

```
        fclose(in);
        return cgiEnvironmentIO;
}

static int cgiReadString(FILE *in, char **s) {
        int len;
        cgiReadInt(in, &len);
        *s = (char *) malloc(len + 1);
        if (!(*s)) {
                return 0;
        }
        if (fread(*s, 1, len, in) != len) {
                return 0;
        }
        (*s)[len] = '\0';
        return 1;
}

static int cgiReadInt(FILE *out, int *i) {
        if (!fread(i, sizeof(int), 1, out)) {
                return 0;
        }
        return 1;
}

static int cgiStrEqNc(char *s1, char *s2) {
        while(1) {
                if (!(*s1)) {
                        if (!(*s2)) {
                                return 1;
                        } else {
                                return 0;
                        }
                } else if (!(*s2)) {
                        return 0;
                }
                if (isalpha(*s1)) {
                        if (tolower(*s1) != tolower(*s2)) {
                                return 0;
                        }
                } else if ((*s1) != (*s2)) {
                        return 0;
                }
                s1++;
                s2++;
        }
```

```
}

static char *cgiFindTarget = 0;
static cgiFormEntry *cgiFindPos = 0;

static cgiFormEntry *cgiFormEntryFindFirst(char *name) {
        cgiFindTarget = name;
        cgiFindPos = cgiFormEntryFirst;
        return cgiFormEntryFindNext();
}

static cgiFormEntry *cgiFormEntryFindNext() {
        while (cgiFindPos) {
                cgiFormEntry *c = cgiFindPos;
                cgiFindPos = c->next;
                if (!strcmp(c -> attr, cgiFindTarget)) {
                        return c;
                }
        }
        return 0;
}

static int cgiFirstNonspaceChar(char *s) {
        int len = strspn(s, " \n\r\t");
        return s[len];
}

void cgiStringArrayFree(char **stringArray) {
        char *p;
        p = *stringArray;
        while (p) {
                free(p);
                stringArray++;
                p = *stringArray;
        }
}
```

File: listen.c

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <linux/in.h>
#include <netdb.h>


/*
Set captionsocket in BSD services file to a socketnumber
9999 for example
*/

/* Relay across socket to Manfred's access,
could be simple socket, httpd on socket 80,
or even fake an odbc!!! since we need only select!!! */


struct sockaddr_in theaddr = { AF_INET };

socket_listen()
{
        struct servent *serv, *getservbyname();
        struct hostent *host, *gethostbyname();
        int ss, ns, length;
        char ipbuf[44];

        if( (ss=socket(AF_INET, SOCK_STREAM, 0)) < 0 )
        {
                puts("\nsocket() failed.");
                return(0);
        }

        /* look up the address of the local host */
        host = gethostbyname("localhost");
        if( host == (struct hostent *)0 )
        {
                puts("\nFailed to get host ip address for this name: \"caption\"");
                return(0);
        }
        bcopy(host->h_addr, (char *)&theaddr, host->h_length);

        theaddr.sin_family = AF_INET;
        length = sizeof(theaddr);
        if (getsockname(ss, (struct sockaddr *)&theaddr, &length) == -1)
        {
                close(ss);
                puts("Bad address for socket.");
                return(0);
        }

        if( (serv = getservbyname("captionsocket","tcp")) == 0 )
        {
                close(ss);
                puts("/etc/services does not have socket caption registered.");
                return(0);
        }
        /*theaddr.sin_port = 0;*/
        theaddr.sin_port = serv->s_port;

        if( bind(ss, (struct sockaddr *)&theaddr, sizeof(theaddr)) < 0)
        {
                puts("\nbind() failed.");
                close(ss);
                return(0);
        }

        if( listen(ss,(int)1) < 0 )
        {
                puts("\nlisten() failed.");
                return(0);
        }

        if( (ns=accept(ss, 0, 0)) < 0 )
        {
                puts("\naccept() failed.");
                return(0);
        }
        close(ss);

        return(ns);
}
```

File: Makefile

```
#You may wish to add -DNO_UNISTD and -DNO_SYSTEM to CFLAGS if
#your system lacks the /usr/include/unistd.h header file
#or the system() function, respectively.

CFLAGS      = -O -DHDUU -DAUTOCONNECT
LDFLAGS     = -s


$(NATEXES):$$@.c
        $(CC) $(CFLAGS) -o $@ $? $(LDFLAGS)

all:  captionsock

captionsock: captionsock.o
        gcc captionsock.o -o captionsock
```